Adaptive Pixel Deviation Absorption Technique for Efficient Video Surveillance using Deep Convolutional Neural Networks

K. Lokesh
Department of Computer Science and Engineering, School of Computing, College of Engineering and Technology, SRM Institute of Science and Technology, Kattankulathur, Chengalpattu, TamilNadu, 603 203, India
lk9977@srmist.edu.in

M. Baskar
Department of Computing Technologies, School of Computing, College of Engineering and Technology, SRM Institute of Science and Technology, Kattankulathur, Chengalpattu, TamilNadu, 603 203, India
baashkarcse@gmail.com (corresponding author)

Received: 13 December 2024 | Revised: 9 January 2025 | Accepted: 12 January 2025

 DOI: https://doi.org/10.48084/etasr.9935

ABSTRACT

Monitoring human activity in industries is a great challenge and numerous methods use various features, such as sketch, position, color, and shape features. However, these methods do not achieve the expected accuracy in classifying the activity of people in the environment. This study presents an Adaptive Pixel Deviation Approximation with Deep Convolutional Neural Networks (APDA-DCNN) model to increase classification accuracy. The method starts with local feature-approximation-based normalization of video frames. Then, global value segmentation is used to group the features of the frame. From the image segmented, the human features along with texture and region pixel deviation features are extracted. The APDA-DCNN model trains the CNN model to convolve the texture features into one-dimensional features by convolving in two layers. The output layer neurons estimate Texture Similarity (TS), Sketch Level Similarity (SLS,) and Pixel Deviation Similarity (PDS) against various classes. Using the values of TS and PDS, the model estimates the Activity Weight (AW) against various classes to select the most dominant. The APDA-DCNN model increases the accuracy of activity classification to achieve higher video surveillance performance.

Keywords-video surveillance; activity classification; APDA-DCNN; TS; PDS; AW

I. INTRODUCTION

The recent increase in social violence has led the industrial sector to enforce video surveillance to monitor and track its employees in work settings. This has been identified as a very important event in organizations and supports the enforcement of industrial security. Tracking and monitoring people in an organization using human intervention is not enough, making it necessary to monitor people in an automated way. By implementing such automated tracking systems, incidents can be instantly identified, and alerts can be generated to stimulate security. This could be enforced by adapting video surveillance with automated activity classification models. Activity classification is the basic stage of any video surveillance system, performed by processing video frames captured through surveillance cameras. Image processing becomes the dominant entity in this case, supporting the identification of human objects and extracting required features from the image. When identifying and extracting human objects, their features can be used for activity classification. Many methods have been proposed for activity classification, varying in terms of features being used and the kind of similarity measure or method being adopted. For example, machine learning methods are more popular in this problem, including particle swarm optimization, Naïve Bayes (NB), Support Vector Machine (SVM), Decision Tree (DT), and Artificial Neural Networks (ANNs). However, machine learning approaches are not suitable for this problem because they are poor at handling larger volumes of data, which is essential to classify activities. Thus, Deep Learning (DL) models have been proposed for this task, including Convolutional Neural Networks (CNNs), LSTM, Generative Adversarial Networks (GANs), and so on.

In terms of DL, CNNs are the most popular approach, since they are capable of reducing feature dimensions without losing information, especially when handling huge volumes of data. This study proposes an Adaptive Pixel Deviation Absorption Technique using Deep CNN (APDA-DCNN) model for this

purpose. Instead of only focusing on texture features, the model uses sketch and pixel deviation features in various regions. By adopting such deviation features, the activity can be classified more efficiently. To achieve this, the model estimates different measures, such as Texture Similarity (TS), Sketch Level Similarity (SLS), and Pixel Deviation Similarity (PDS). TS is measured according to the pixel similarity between textures. SLS is measured according to the similarity among the sketches and PDS is measured according to the pixel deviation in each region of the sketches. Using these measures, the model can estimate the activity weights for various classes to perform the classification.

II. RELATED WORKS

In [1], a Surveillance Video Quality Analysis (SVQA) approach was introduced, presenting Distorted Face Verification (DFV-SVQA) and Distorted Face Identification (DFI-SVQA). This approach used a CNN network for face recognition to generate a surveillance dataset. In [2], a spatiotemporal GAN was proposed, which detected the dynamic texture content on the encoder side. This method used GAN to generate textures, and neurons measured the correlation on spatial and temporal neighbors to perform classification. In [3], a Multi-Level Video Security (MuLViS)-based surveillance system was proposed, which used a security ontology to allow automatic selection of cameras and restrict unauthorized access. In [4], a block-level Background Reference Frame (BRF) approach was presented to handle the redundancy issue and generate a Foreground Reference Frame (FRF) according to a Surveillance Prediction Generative Adversarial Network (SP-GAN) scheme with the use of previously reconstructed frames. The detection was performed by optical flow. In [5], a cloud-based video surveillance system was presented for different attacks, examining different vulnerabilities, threats, and attack classification. In [6], a novel video compression scheme was presented, using adaptive background updating and interpolation by sharing background information with adjacent frames. In [7], an IoT-based architecture was presented, named IoVT-VSS (Internet of Video Things Video Surveillance System), which transmitted video frames through IoT devices to carry out effective surveillance. In [8], a video Synthetic Aperture Radar (Video-SAR) was proposed to support video surveillance in 24 hours, which produced a sequence of videos to monitor day and night conditions. In [9], a Scene Adaptive Octree-based model (SSOcT) was introduced to analyze scene characteristics and extract spatiotemporal structures from videos. The octree-based algorithm was used for classification. In [10], a detailed analysis of a trajectory-based surveillance model was presented to cluster videos and generate synopses. In [11], a hashing- and steganography-based approach was presented for video surveillance, storing encryption keys in a hardware wallet and evidence using steganography. In [12], an efficient edge computing-based Video Usefulness (VU) approach was presented to determine online failures. In [13], a pedestrian detection algorithm, named robustness quadrangle, was presented, along with a large-scale Distorted Surveillance Video Dataset (DSurVD). This dataset included different high-quality video sequences for pedestrian detection. In [14], a low-power surveillance video coding approach was presented, which used inter-motion estimation to segment the input video Frame Memory Compression (FMC) for segmentation and video surveillance. The ViTrack multi-video tracking framework [15] used spatial and temporal features in two different layers. Based on the features, a set of relations was derived to perform classification with the Markov model. In [16], a detailed analysis of video surveillance schemes was discussed along with the OpenCV library and moving target detection algorithms for efficient surveillance.

In [17], an improved energy minimization scheme was proposed, which used SA and JAYA algorithms for faster convergence. In [18], a Motion From Memory (MFM) method was presented, which extracted features and kept track of different sequences of motion frames. The features extracted were used to train a lightweight MobileNet-based Faster RCNN detector for classification. In [19], a robust privacy-preserving motion detection scheme was presented, which included multiple object-tracking schemes for encrypted surveillance video bitstreams. This method used adaptive clustering and segmentation schemes to improve feature extraction and video surveillance accuracy. In [20], a reversible face de-identification scheme was presented for video surveillance, using landmarks and CGAN to extract features from faces and perform detection. In [21], a sparsity and low-rank contextual regularization scheme was proposed to detect moving objects in multi-scenario surveillance data. In [22], an adversarial-oriented DL framework was proposed, which analyzed the performance of face mask surveillance against adversarial threats by adapting the ShuffleNet V1 transfer learning algorithm for face mask detection. In [23], the relationship between objects and actors was analyzed using 3D convolutions and spatiotemporal data to detect violent acts. In [24], a Multi-Objective Hybrid Aquila optimizer with Simulated Annealing (MOHASA) was proposed to handle the problem of missing views and portions by generating a video synopsis with spatial and temporal features to perform video surveillance. In [25], a multi-attention DL model was proposed, which adapted the EfficientNet-B0 architecture by integrating a Convolutional Block Attention Module (CBAM) to maximize the performance of feature extraction in detecting anomalies. In [26], a CNN-based model was presented for privacy protection with behavior recognition, which integrated multi-scale feature fusion, spatiotemporal attention, and LSTM to maximize performance. In [27], an intelligent surveillance framework for Artificial Intelligence of Things, named Ancilia, focused on ensuring privacy in video surveillance.

III. ADAPTIVE PIXEL DEVIATION ABSORPTION VIDEO SURVEILLANCE USING DEEP CNN (APDA-DCNN)

The APDA-DCNN model starts with local feature approximation-based normalization of video frames. Segmentation is carried out using global value segmentation to cluster the features of the frame. From the image segmented, human features are extracted along with texture and region pixel deviation features. The APDA-DCNN model trains the CNN model to convolve the features of texture into one-dimensional features by convolving in two layers. The output layer neurons estimate Texture Similarity (TS), Sketch Level

Similarity (SLS), and Pixel Deviation Similarity (PDS) against various classes of features. Using the value of TS and PDS, the model estimates the Activity Weight (AW) against various classes to select the most dominant one. Figure 1 shows the working diagram of the APDA-DCNN model.

A. Local Feature Approximation

The local feature approximation algorithm works with the features of neighboring pixels to normalize itself. The method traverses each pixel and collects both color and gray features. The number of neighbor pixels considered is up to 3 levels. For each pixel, the method identifies the available 3-level pixels and collects the color and gray features. The Local Intensity Mean (LIM) and Local Gray Mean (LGM) are calculated to normalize the pixel value. According to these two values, the method decides whether the pixel should be normalized or not by calculating the Pixel Deviation Value (PDV). According to the value of PDV, the method adjusts the pixel value to normalize the pixel and enhance the quality of the image. The local feature approximation algorithm calculates LIM and LGM values to measure PDV, normalize image pixels, and improve quality.

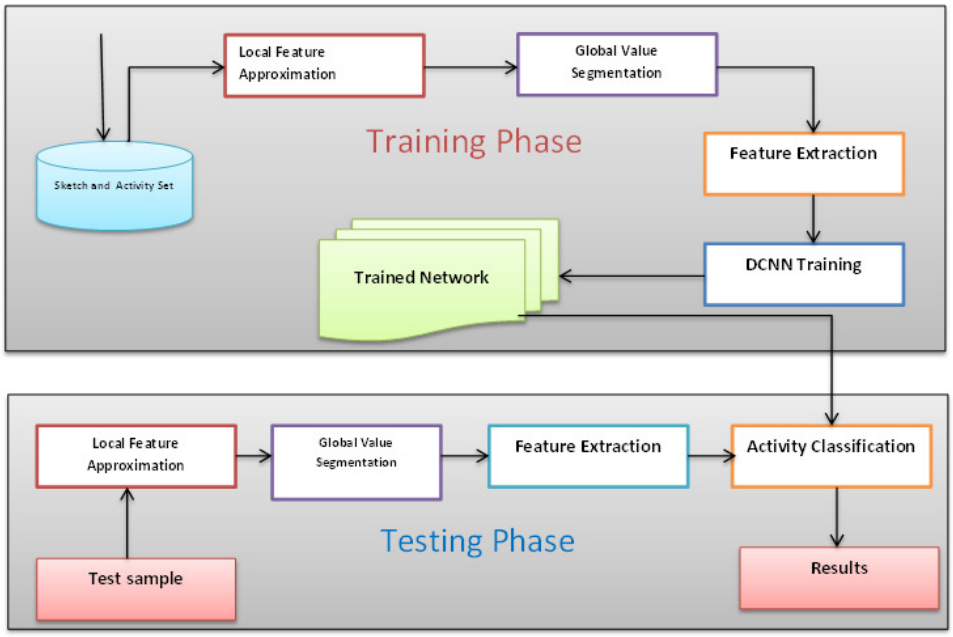

Fig. 1. Functional diagram of the APDA-DCNN model.

Algorithm 1: Local Feature Approximation
Input: Frame Image F_{img}
Output: Enhanced image E_{img}
Begin
Fetch F_{img}
Initialize $k = 3$
For each pixel p
  Pixel group pg = Collect k neighbors in all directions
  Calculate $LIM = \frac{\sum_{i=1}^{size(Pg)} pg(i).r}{size(Pg)}$
  Calculate $LGM = \frac{\sum_{i=1}^{size(Pg)} pg(i)(grayvalue)}{size(Pg)}$
  Calculate $PDV = LIM \times LGM$
  If $(p.intensity \times p.grayvalue) \geq (\frac{2}{3} \times PDV)$ then
    $F_{img}(p).Intensity = p.intensity + (\frac{1}{6} \times LIM)$
    $Fimg(p).grayvalue = p.grayvalue + \frac{1}{10} \times LGM$
  EndIf
EndFor
$F_{img} = E_{img}$
EndAlgorithm

B. Global Value Segmentation

The global value segmentation scheme uses the grayscale values of the entire image as the key to segmenting it. The method collects the entire gray feature and computes the Global Gray Mean (GGM). Accordingly, it generates the histogram of features and selects a subset of gray values from both sides of the GGM value. According to the subset of gray values selected, the method generates clusters. By traversing through each pixel, the method estimates the Object Gray Mean Distance (OGMD) toward various clusters of pixels. Based on the OGDM value, the method identifies the group of pixels and the index to them. Finally, the grouped pixels are mapped to produce the segmented image. The global value segmentation algorithm estimates the OGMD value to group the pixels of the object to perform segmentation.

Algorithm2 : Global Value Segmentation
Input: Enhanced Image E_{img}

Output: Segmented image S_{img}, Object Ob_s
Begin
Fetch E_{img}
$size(E_{img})$
Gray set $Gs = GrayScale(E_{img}(i))$
$i = 1$
$size(Gs)$
Calculate $GGM = \frac{\sum Gs(i)}{Size(Gs)}$
$Hiss = Histogram\,(E_{img})$
Initialize cluster set with a size of 5.
Selective Gray Scale set:
$Sgss = Sgss(1) = GGM$
Identify the top two occurring gray values from both sides of GGM and add them to the set $Sgss$
For each pixel p
  For each cluster c
    Calculate $OGMD = \frac{\sum Dist(C(i).grayvalue,p.value)}{size(C)}$
  EndFor
  Choose the class with the least OGMD and index the pixel to the cluster.
EndFor
Map all the cluster pixels to the S_{img}.
Ob_s = Extract the objects from the segmented image
EndAlgorithm

IV. FEATURE EXTRACTION

The proposed model extracts the texture of human objects, shape features, and pixel deviation features from the image. To achieve this, the method computes Human Perception Measure (HPM) for various activity class objects. HPM is measured based on the number of pixels that are correlated with the pixels of the object in the class. Based on the HPM value, the method identifies the human object and extracts texture and sketch features. Based on the sketch features extracted, the pixel deviation features are extracted according to the number of pixels present in each region. The extracted features are converted into a feature vector for activity classification. This method extracts texture, sketch, and PDF features from the image to support the training process.

Algorithm 3: Feature Extraction
Input: Object set Ob_s, Activity template set At_s, Segmented image S_{img}
Output: Texture Feature Tf_e, Sketch s, Pixel Deviation Feature PDF
Start
Fetch At_s, Ob_s, S_{img}
For each object o
  For each activity class AC
    For each object Ao
      $size(AC)$
      $size(O)$
      $size(AC(j))$
      $HPM = \frac{\sum_{k=1}^{j=1} Sum(Count(AC(j)(k)==O(i)) / size(Ac(j)))}{Size(Ac)}$
    EndFor
    $HPM = \sum HPM / Size(AC)$
  EndFor
EndFor
Object O = Choose the object with the highest HPM value
Texture Tex = Extract the texture of the object
Sketch S = Extract the object sketch
$size(Region)$
$PDF = Count(Number\ of\ pixels\ in\ each\ region)$
$i = 1$
EndAlgorithm

A. DCNN Training

The model loads the dataset and applies local feature approximation on each image. The normalized image is segmented using global value segmentation. The texture, sketch, and pixel deviation features are extracted from the segmented image. These features are used to train the convolution layer, which convolves the texture and sketch in two layers and applies pooling at each intermediate layer. The output layer calculates TS, SLS, and PDS against various classes of features.

B. Activity Classification

The APDA-DCNN model applies local feature normalization to enhance image quality. Segmentation is performed using global values. The extracted features are passed to the pre-trained model, where the convolution neurons convolve the texture and sketch features to reduce size, and output layer neurons estimate TS, SLS, and PDS against various classes of features. Using the value of TS and PDS, the model estimates the AW against various classes to select the most dominant one as the result. The proposed model estimates AW against various classes and populates a single class with the maximum AW.

Algorithm 4: Activity Classification
Input: DCNN, Test image T_{img}
Output: Activity Class AC
Begin
Read DCNN and T_{img}
  N_{img} = Local feature approximation(T_{img})
  S_{img} = Global value segmentation(N_{img})
[Texture, Sketch, PDF] =
  Feature_Extraction (S_{img})
Pass the extracted features through the DCNN
At convolution_layer_1

Convolve texture and sketch features to dimension x.
Apply max_pooling
At convolution_layer_2
Generate LBP feature from texture,
generate region-wise pixel count on sketch
Apply max pooling
With each neuron n

$$Ts = \frac{\sum_{i=1}^{Size(Ts)} TS(i).value == n.LBP}{size(Ts.LBP)}$$

$$SLS = \frac{\sum_{i=1}^{Size(C)} Count(C(i).pixelcount == Test.pixelcount)}{size(C)}$$

$$PDS = \frac{\sum_{i=1}^{4} Count(C(i).PDF == Test.PDF(i))}{4}$$

End
Calculate Activity Weight AW

$$AW = \frac{\sum Ts}{\sum SLS} \times \sum PDS$$

End
Class AC = Populate class with maximum AW
EndAlgorithm

V. RESULTS AND DISCUSSION:

The APDA-DCNN model was coded in Matlab. The efficacy of the model was evaluated using different activity sets using the Kaggle DCSASS dataset [28], which contains 13 activity classes. The activity set contained Abuse, Arrest, Arson, Assault, Accident, Burglary, Explosion, Fighting, Robbery, Shooting, Stealing, Shoplifting, and Vandalism.

TABLE I. EXPERIMENTAL SETUP

Factor	Feature_value
Activities	13
Dataset	Kaggle (DCSASS) [28]
Dataset Size	16853
Tool	Matlab
No of Users	500

The dataset was split into three incremental activity sets to evaluate performance, using training/test ratios of 60:40, 70:30, and 80:20. Accuracy was measured in three test cases, considering the first four activities, the first eight activities, and all activities, with different training/testing split ratios. Table II shows the accuracy of the model in each case.

TABLE II. ACTIVITY CLASSIFICATION ACCURACY

Accuracy % vs Activities			
Train/test %	**4 Activities**	**8 Activities**	**13 Activities**
80:20	83	89	98
70:30	81	84	93
60:40	78	83	87

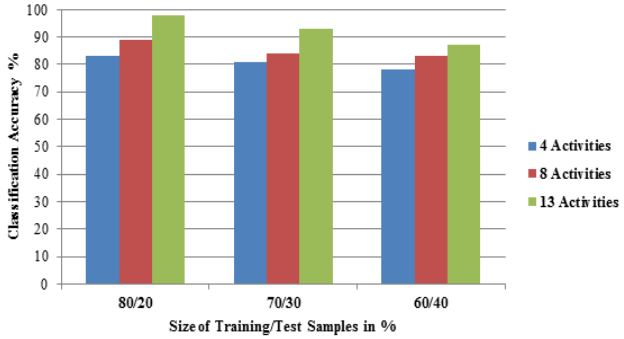

Fig. 2. Accuracy on activity classification.

Table III and Figure 3 show the false rate ratios in activity classification.

TABLE III. FALSE RATE RATIOS

False Ratio % vs Activities			
Train/test %	**4 Activities**	**8 Activities**	**13 Activities**
80:20	17	11	2
70:30	19	16	7
60:40	22	17	13

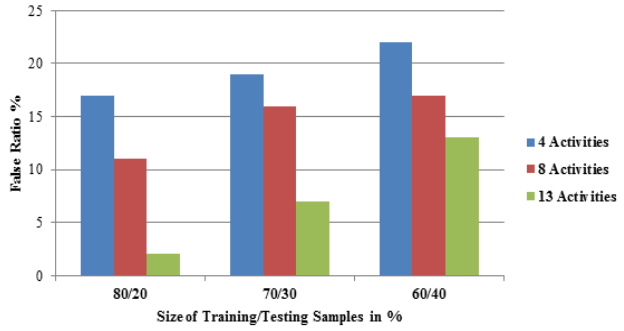

Fig. 3. False ratios in activity classification.

Table IV and Figure 4 show the classification time for different training/testing ratios.

TABLE IV. TIME COMPLEXITY

Time Complexity (s) vs Activities			
Activities	**4 Activities**	**8 Activities**	**13 Activities**
80/20	23	32	48
70/30	65	74	85
60/40	54	67	78

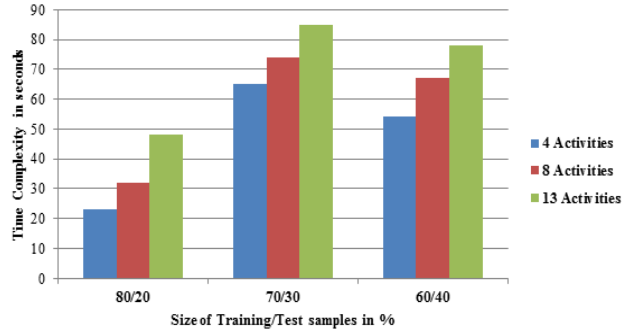

Fig. 4. Time complexity results.

VI. CONCLUSION

The proposed APDA-DCNN model starts with local feature approximation-based normalization of video frames. Then, segmentation is performed using global value segmentation to group the features of the frame. The segmented image is used to extract the texture, region, and pixel deviation features. This approach improves segmentation and the efficiency of feature extraction. The CNN model convolves the texture features into one-dimensional features. The output layer neurons estimate texture similarity, sketch level similarity, and pixel deviation similarity against various classes of features. The model estimates the activity weight against various classes to select the most dominant one. Considering pixel deviation and absorption in segmentation, the model stimulates the accuracy of activity detection. The proposed model improves the performance of video surveillance and activity classification, reaching an accuracy of 98%.

REFERENCES

[1] W. Heng, T. Jiang, and W. Gao, "How to Assess the Quality of Compressed Surveillance Videos Using Face Recognition," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 29, no. 8, pp. 2229–2243, Aug. 2019, https://doi.org/10.1109/TCSVT.2018.2866701.

[2] K. Yang, D. Liu, Z. Chen, F. Wu, and W. Li, "Spatiotemporal Generative Adversarial Network-Based Dynamic Texture Synthesis for Surveillance Video Coding," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 32, no. 1, pp. 359–373, Jan. 2022, https://doi.org/10.1109/TCSVT.2021.3061153.

[3] A. Shifa *et al.*, "MuLViS: Multi-Level Encryption Based Security System for Surveillance Videos," *IEEE Access*, vol. 8, pp. 177131–177155, 2020, https://doi.org/10.1109/ACCESS.2020.3024926.

[4] L. Zhao, S. Wang, S. Wang, Y. Ye, S. Ma, and W. Gao, "Enhanced Surveillance Video Compression With Dual Reference Frames Generation," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 32, no. 3, pp. 1592–1606, Mar. 2022, https://doi.org/10.1109/TCSVT.2021.3073114.

[5] D. Aklamati, B. Abdus-Shakur, and T. Kacem, "Security Analysis of AWS-based Video Surveillance Systems," in *2021 International Conference on Engineering and Emerging Technologies (ICEET)*, Istanbul, Turkey, Oct. 2021, pp. 1–6, https://doi.org/10.1109/ICEET53442.2021.9659574.

[6] L. Wu, K. Huang, H. Shen, and L. Gao, "Foreground-Background Parallel Compression With Residual Encoding for Surveillance Video," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 31, no. 7, pp. 2711–2724, Jul. 2021, https://doi.org/10.1109/TCSVT.2020.3027741.

[7] T. Sultana and K. A. Wahid, "Choice of Application Layer Protocols for Next Generation Video Surveillance Using Internet of Video Things," *IEEE Access*, vol. 7, pp. 41607–41624, 2019, https://doi.org/10.1109/ACCESS.2019.2907525.

[8] M. R. Khosravi and S. Samadi, "Mobile multimedia computing in cyber-physical surveillance services through UAV-borne Video-SAR: A taxonomy of intelligent data processing for IoMT-enabled radar sensor networks," *Tsinghua Science and Technology*, vol. 27, no. 2, pp. 288–302, Apr. 2022, https://doi.org/10.26599/TST.2021.9010013.

[9] Y. Yang, H. Kim, H. Choi, S. Chae, and I. J. Kim, "Scene Adaptive Online Surveillance Video Synopsis via Dynamic Tube Rearrangement Using Octree," *IEEE Transactions on Image Processing*, vol. 30, pp. 8318–8331, 2021, https://doi.org/10.1109/TIP.2021.3114986.

[10] S. A. Ahmed, D. P. Dogra, S. Kar, and P. P. Roy, "Trajectory-Based Surveillance Analysis: A Survey," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 29, no. 7, pp. 1985–1997, Jul. 2019, https://doi.org/10.1109/TCSVT.2018.2857489.

[11] N. Kanwal *et al.*, "Preserving Chain-of-Evidence in Surveillance Videos for Authentication and Trust-Enabled Sharing," *IEEE Access*, vol. 8, pp. 153413–153424, 2020, https://doi.org/10.1109/ACCESS.2020.3016211.

[12] H. Sun, W. Shi, X. Liang, and Y. Yu, "VU: Edge Computing-Enabled Video Usefulness Detection and its Application in Large-Scale Video Surveillance Systems," *IEEE Internet of Things Journal*, vol. 7, no. 2, pp. 800–817, Feb. 2020, https://doi.org/10.1109/JIOT.2019.2936504.

[13] Y. Fang, G. Ding, Y. Yuan, W. Lin, and H. Liu, "Robustness Analysis of Pedestrian Detectors for Surveillance," *IEEE Access*, vol. 6, pp. 28890–28902, 2018, https://doi.org/10.1109/ACCESS.2018.2840329.

[14] H. Kim and H.-J. Lee, "A low-power surveillance video coding system with early background subtraction and adaptive frame memory compression," *IEEE Transactions on Consumer Electronics*, vol. 63, no. 4, pp. 359–367, Nov. 2017, https://doi.org/10.1109/TCE.2017.015073.

[15] L. Cheng, J. Wang, and Y. Li, "ViTrack: Efficient Tracking on the Edge for Commodity Video Surveillance Systems," *IEEE Transactions on Parallel and Distributed Systems*, vol. 33, no. 3, pp. 723–735, Mar. 2022, https://doi.org/10.1109/TPDS.2021.3081254.

[16] J. Huang, A. Huang, and L. Wang, "Intelligent Video Surveillance of Tourist Attractions Based on Virtual Reality Technology," *IEEE Access*, vol. 8, pp. 159220–159233, 2020, https://doi.org/10.1109/ACCESS.2020.3020637.

[17] S. Ghatak, S. Rup, B. Majhi, and M. N. S. Swamy, "HSAJAYA: An Improved Optimization Scheme for Consumer Surveillance Video Synopsis Generation," *IEEE Transactions on Consumer Electronics*, vol. 66, no. 2, pp. 144–152, May 2020, https://doi.org/10.1109/TCE.2020.2981829.

[18] W. Liu, S. Liao, and W. Hu, "Perceiving Motion From Dynamic Memory for Vehicle Detection in Surveillance Videos," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 29, no. 12, pp. 3558–3567, Dec. 2019, https://doi.org/10.1109/TCSVT.2019.2906195.

[19] X. Tian, P. Zheng, and J. Huang, "Robust Privacy-Preserving Motion Detection and Object Tracking in Encrypted Streaming Video," *IEEE Transactions on Information Forensics and Security*, vol. 16, pp. 5381–5396, 2021, https://doi.org/10.1109/TIFS.2021.3128817.

[20] H. Proenca, "The UU-Net: Reversible Face De-Identification for Visual Surveillance Video Footage," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 32, no. 2, pp. 496–509, Feb. 2022, https://doi.org/10.1109/TCSVT.2021.3066054.

[21] B. H. Chen, L. F. Shi, and X. Ke, "A Robust Moving Object Detection in Multi-Scenario Big Data for Video Surveillance," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 29, no. 4, pp. 982–995, Apr. 2019, https://doi.org/10.1109/TCSVT.2018.2828606.

[22] G. R. Panigrahi, P. K. Sethy, S. K. Behera, M. Gupta, F. A. Alenizi, and A. Nanthaamornphong, "Enhancing Security in Real-Time Video Surveillance: A Deep Learning-Based Remedial Approach for Adversarial Attack Mitigation," *IEEE Access*, vol. 12, pp. 88913–88926, 2024, https://doi.org/10.1109/ACCESS.2024.3418614.

[23] V. D. Huszár, V. K. Adhikarla, I. Négyesi, and C. Krasznay, "Toward Fast and Accurate Violence Detection for Automated Video Surveillance Applications," *IEEE Access*, vol. 11, pp. 18772–18793, 2023, https://doi.org/10.1109/ACCESS.2023.3245521.

[24] S. Priyadharshini and A. Mahapatra, "MOHASA: A Dynamic Video Synopsis Approach for Consumer-Based Spherical Surveillance Video," *IEEE Transactions on Consumer Electronics*, vol. 70, no. 1, pp. 290–298, Feb. 2024, https://doi.org/10.1109/TCE.2023.3324712.

[25] S. Ul Amin, M. Sibtain Abbas, B. Kim, Y. Jung, and S. Seo, "Enhanced Anomaly Detection in Pandemic Surveillance Videos: An Attention Approach With EfficientNet-B0 and CBAM Integration," *IEEE Access*, vol. 12, pp. 162697–162712, 2024, https://doi.org/10.1109/ACCESS.2024.3488797.

[26] W. Yuan, "Enhancing Video Surveillance and Behavior Recognition With Deep Learning While Ensuring Privacy Protection," *IEEE Access*, vol. 12, pp. 157466–157476, 2024, https://doi.org/10.1109/ACCESS.2024.3486051.

[27] A. D. Pazho *et al.*, "Ancilia: Scalable Intelligent Video Surveillance for the Artificial Intelligence of Things," *IEEE Internet of Things Journal*,

vol. 10, no. 17, pp. 14940–14951, Sep. 2023, https://doi.org/10.1109/JIOT.2023.3263725.

[28] "DCSASS Dataset." Kaggle, [Online]. Available: https://www.kaggle.com/datasets/mateohervas/dcsass-dataset.