An Efficient Unusual Event Tracking in Video Sequence using Block Shift Feature Algorithm

Karanam Sunil Kumar[1]
Assistant Professor
Department of Computer Science and Engineering
RNS Institute of Technology
Bangalore, India

Dr. N P Kavya[2]
Professor
Department of Computer Science and Engineering
RNS Institute of Technology
Bangalore, India

***Abstract*—The area of video technology is rapidly growing owing to advancements in intelligent video systems in sensor operations, higher bandwidth capacity, storage, and high-resolution displays. This led to the proliferation of video-based computing modeling to perform specific tasks on video sequences to gain more insight from the data. Visual tracking of events is a core component in video visual surveillance systems that classify and track moving objects to describe their behavioral aspects. The prime motive behind intelligent video systems is to perform efficient video analytics to meet the specific requirements of the user/use-cases. It involves a self-directed paradigm to understand event sequences, reducing the computational burden of characterizing the activities. The study incorporates a block-shift feature algorithm and introduces a novel computational research method for unusual event tracking in video sequences. The formulated approach employs a framework combining operational blocks to compute sequential operations such as block-matching from the dictionary of motion estimations. Before applying the learning model, the subsequent analysis procedure adds feature lexicon and dominant attributes to make the execution computationally efficient. Further, it uses a sparse-non negative factorization approach to organize the informative details into k possible finite clusters. The event detection outcome from the training datasets of video sequences shows better experimental results than the traditional highly cited related approach of unusual object detection and tracking.**

Keywords*—Object detection; tracking; learning models; video sequence analysis

I. Introduction

The ever-increasing surveillance process adopted in every walk of life leads to the video sequence databases' reposit. If these databases are not analyzed, it is just a kind of dead data. The analysis of textual data is quite simple as the basic terms construct it. In contrast, the video sequences contain various sensitive pattern features and information. Its interpretation requires suitable and efficient algorithms to deal with its analysis so that multiple applications can be built based on this analysis, and real-time decisions can be taken. The advancement of the computing platforms with the CPU and GPU functioning provides the right ecosystem to explore the possibilities of automatically tracking an object in the video sequences. In the computer vision domain, detecting and tracking the objects within video sequences plays an essential role in many applications [1]. Some of the examples of such applications include i) event tracking in video surveillance [2], ii) dynamic road traffic management [3], iii) intruder tracking [4], and iv) robotics vision [5]. The individual image sequences in the video are popularly known as a frame in video processing. These frames contain both the static and moving object into it. The structure's static part is termed the background, whereas the moving objects are known as the foreground [6]. The task of tracking objects in the video frames or sequences includes detecting moving objects of interest, then classifying them and following them from frame to frame.

In the process of object detection, the pixels of the object in interest are clustered, which is generally performed by various methods, including i) frame differences, ii) subtraction of the background, and iii) optical flow computation [7]. The object classification occurs only after the object detection based on the high-level features, including one or combinations of color, shape, texture, or sometimes motion. However, the tracking of an object is to gain specific information on the object utilizing its orientation, activity, occlusion, etc., using three popular approaches i) silhouette [8], ii) kernel [9], and iii) point-based tracking [10]. Point-Based Tracking (PBT) is significant for tracking small objects in the video sequences; however, it does not perform well in occlusion and provides false detection [11]. The PBT is broadly classified into i) Kalman filtering (KF) [12] and ii) Particle filtering (PF) [13]. The typical working process of the KF is to perform prediction of the current state variables and correction recursively. Since KF handles the noisy input data in recursion, it can track single objects in a real-time scenario [14]. Though the PF also performs the prediction and correction similar to the KF, it overcomes the state variable's approximation constraints by re-sampling.

PF generates many possible models for the current variable before moving to the following state variable. The PF exploits feature sets like {color, contour, texture}[15]. The Kernel-Based Tracking (KBT) uses shapes like rectangles or ellipses to keep the object of interest outside the shape and background to detect rigid objects. The typical classification of the KBT includes i) Simple Template Matching (STM), ii) Mean Shift (MS), iii) Support Vector Machine (SVM), and iv) Layer Matching (LM) [16]. The STM is applicable for tracking a single object in partial occlusion conditions. It verifies the video sequences with a reference frame, and only motion transformation is possible in this method. Simultaneously, the MS method uses chamfer distance transformation, the Bhattacharya coefficient for distance, and color distribution

transformation among the region of interest in windows to windows to improve the accuracy. However, it cannot track high-speed moving objects and only single object tracking [17]. However, the SVM-based method tracking the object in video sequences tracks only the positive samples. This method also handles tracking a single object and limits its applicability to partial occlusion [18]. The LM- method is used along with the KB method to track multiple objects, even in the complete occlusion condition [19]. The complex object containing a composite shape is well followed using the Silhouette-based method, which is further categorized into i) contour tracking (CT) [20] and ii) shape matching (SM) [21]. In CT, the object's motion and shape are considered using a state-space model. Then, the contour energy is minimized using gradient descent for computing the next frame's contour in iteration from the previous frames. This method helps track objects with irregular shapes. However, the shape-based process is the same as SMM without classifying the objects, and it uses the edge templates and occlusion utilizing Hough transformations.

This paper proposes a moving object tracking with satisfying background conditions using a block-matching method from the dictionary of the motion estimations. The contribution and potential benefits of proposed approach are as follows:

- The presented model introduces a learning-based approach, initially incorporates-video sequence exploration, followed by block-wise feature lexicon extraction, updates the dictionary of feature lexicon with dominant attributes.
- The system also includes a novel feature engineering schema and workflow modeling to compute feature elements with lexicon vector.
- The study model discusses about a novel sparse non-negative factorization (S-NNF) technique that factorizes data organization into k possible finite clusters.
- The study further applies a simplified learning-based systematic approach to faster tracking moving objects from the video feed of frame sequence.
- The system is analytically represented with numerical analysis, which subjects to the dictionary-based learning operation towards tracking the moving object anomaly.
- The study also simulates the devised numerical models with the systematic workflow execution for evaluating the performance. The outcome clearly shows the effectiveness of the block-shift feature algorithm in terms of different classification parameters.

All the above mentioned contribution are implemented in a sequential way. The organization of the paper is as follows: Section II discusses about existing methodologies while its identified problems are briefed in Section III. Section IV discusses about the system model while result discussion is carried out in Section V. The summary of the paper in form of conclusion is briefed in Section VI

II. Review of Literature

Various strategies are evolving to improve the video tracking system's operation and performance. To develop an effective video tracking system, the existing approaches consider selecting multiple parameters, e.g., points, primitive geometric shape, contour, Silhouette of an object, articulated shape, and skeletal models. Such conventional approaches are usually modeled using standard features, e.g., color, edges, optical flow, and texture. However, the video tracking system has recently introduced various unique techniques. The existing approach offers importance to extracting the semantic factor associated with capturing essential features for performing the track. Boukhers et al. [22] have discussed a probability-based model for obtaining trajectories of a three-dimensional object from two-dimensional video feeds. The model can estimate the object depth from the calculated focal length. The technique also uses a particle filtering method based on the Markov chain Monte Carlo process to stabilize the video better feeds with reduced time consumption for detecting an object. However, the majority of the video tracking application emphasizes more on human as an object. The challenge in this field is to differentiate the object from the background based on appearance and color. This issue is sorted out by Damotharasamy [23] by using a sparse representation that is not affected by any variations caused due to illumination. Apart from this, the issue associated with occlusion is addressed using subspace learning that extracts the visual features during the dictionary's updating process.

Further work also performs sophisticated video tracking of moving objects and recognition of specific actions. Studies using a similar methodology were also encountered in the literature using the discrete tracking mechanism by Kong et al. [24]. This approach has considered the localization of coordinates of a moving object using compressive tracking. The study has contributed to refining the scaling factor and recovering the occlusion issue. A convolution network is used in this study for recognizing the action of the moving object based on pre-defined information of the actions and extracts potential features considering the static data of an object.

Existing studies have also witnessed a video tracking system from multi-feeds that offers more elaborated information of the same scene event captured from multiple cameras mounted in different locations. A similar concept has been modeled by Lee et al. [25], where segmentation and tracking multiple objects using dual forms of the feature are applied. Feedback with multi-kernel is used for detecting local objects while performing a video track with a single camera. At the same time, the contextual information and appearance are integrated to carry out multi-camera tracking. The study also used an unsupervised learning method to improve the scalability factor. However, the study is all about tracking a single object. A unique variant of this approach was seen in Liu et al. [26], capable of tracking multiple objects. Independent from any specific object model, this approach can differentiate two similar objects based on the trajectory generation. The system uses a neural network and graphical model for tracking with a correlation between the targets. Another part of the literature on video tracking has been carried out considering detecting abnormalities present within it.

Adopting the Gaussian mixture is proven to improve the video tracking system effectively, as seen in Ratre and Panchapakesan [27]. The model offers a better classification of an object and the decomposition concept's usage using tucker tension. The study also uses cosine similarity to compare the attributes of decomposition considering the mobility-based feature, i.e., speed and shape of an object, to track the event from the video feed. Existing studies also include the text as another representation of an object from the web video for exhibiting the video tracking mechanism. Tian et al. [28] further works in this direction and come up with a Bayesian-based algorithm model for object detection and tracking from a complex video form. A similar kind of tracking of the object is also carried out by Yang et al. [29] by addressing its multi-orientation. The study uses multiple frames using dynamic programming for performing video tracking over various scenes.

Literature has also witnessed the implication of the identification and classification of a different number of objects for a video feed, as seen in the work of Wong et al. [30]. The author has developed a model independent of any apriori information of the object to perform tracking. At the same time, the classification is carried out using the learning approach of the neural network. There are specific unique categories of studies on video tracking. Existing literature has also seen the usage of super-pixels to extract fluctuation of an object's appearance with the monitoring, as seen in the work of Cheng et al. [31]. Using a deep residual network, the classification performance is improved using a correlation filter in tracking. The concept of facial recognition for automated identification of a specific human as an object is presented by Khan et al. [32]. The study also uses location information and time to carry out the tracking process. The adoption of infrared modalities and Red, Green Blue content (RGB) was used to develop a video tracking system for industrial surveillance systems, as explored in Lan et al. [33]. Adopting machine learning has addressed the issues associated with modalities' discrepancies. Another unique implementation is carried out by Liu et al. [34] and Benthem et al. [35]. A stochastic nature grammar is used over a decomposed graph to manage different attributes of the signal feeds. A learning-based approach is delivered for training this grammar model. This work addresses various possibilities of error in object recognition from the video feed. Therefore, multiple methods have been evolved to brief the associated issues in the next section.

III. Research Problem

A review of existing approaches towards improving video tracking performance shows various procedures in recent times and also outlines their limitation factors. However, a closer look into existing systems shows its emphasis on identifying a single mobile object. The studies also offer minor inclusion of contextual attributes connected with objects' mobility patterns. Various studies using trajectories are not meant to evaluate the dynamic environment over the scene. This could be a more significant challenge when tracking an object with dynamic mobility patterns in the crowd. Irrespective of approaches using the extraction of particular objects, the studies lack the consideration of other similar objects moving along with the target object, which could generate a significant number of outliers in its detection process. Although this challenge is somewhat solved using a machine learning-based approach, it should be noted that such systems are computationally complex when it relates to their practical operation of video tracking. Simultaneously, dependencies of trained feeds will also include many resources to store and process them. Such a phenomenon will lag in the bounding box's appearances over the tracked target presence within a scene. Therefore, there is a potential need for research to be carried out to extract contextual information about the target object from the video feed scene to improve accuracy. Learning-based approaches are a potential solution to overcome this challenge; however, such methods also require the smart amendment to balance the spontaneity in tracking performance and computational complexity.

IV. System Model

This study continues our prior works in [36], [37]. The formulated system model consists of various sub-computational units as i) video sequence explorer: where the training and testing data are visualized to get an intuition about the scene, ii) Dictionary feature lexicon blocks: where the dictionary based on the blocks are made, iii) Feature engineering modeling with feature element and lexicon vector computation, followed by iv) Design and development of a cost-effective learning model to estimate learning-based features which help in identifying the unusual moving object from each frame of the input video sequence and v) Exploration of the numerical outcome to justify the performance of the proposed modeling. The consecutive sections illustrate the rational description corresponding to the system modeling concept with mathematical notions.

A. Video Sequence Exploration

The system model provision to select the reposit location (R_l) of the training dataset (D_t), which consists of 'n' video sequences (Vs). The explicit function $f_1([R_l, D_t] \rightarrow :$ Sc[Vs. (N)], the typical elements of the structure set Sc={Na, D, S}, where Na = video sequence name, D=date of creation that may trace the event's date in the surveillance system, and **S**= memory space. The statistical description of the dataset is given in Table I.

The dataset typically consists of '44' independent folders consisting of '8800' video sequences in totality, divided into 77.28 % as training data and 22.72 % as testing data. Few random video sequences from both training data and the testing data are shown below in Table II.

TABLE I. Statistics of the Dataset

Sl. No	Data Description	No of the video sequences
1	Training Data	6800
2	Testing Data	2000

B. Dictionary Feature Lexicon Block

The computational block for obtaining the 'Dictionary of Feature Lexicon' (DFL) takes three input sets {D_t, R_l, Sc[Vs. (n)]}, the detailed operations are given in the algorithm -1.

Algorithm 1: Block wise Feature Lexicon

Input: D_t , R_l , Sc[Vs(n)]
Output:DFL
Process
Start
Initiate: BS← [n x n x n]
F1←Sc[Vs(n)], n=1
B←Convert F1 into a column of block n x n
Create Dictionary Size:
nR←n^3 and nC← q [no. of columns(B)]
oBS← n^2
Cs← with Size [n^3,q , D= $\frac{\lfloor N(Vs)_t \rfloor}{n}$]
Compute: Mean of B as $\vec{M}$

$$V = \frac{B_j - \vec{M}}{\left\| B_j - \vec{M} \right\|}$$

Update DFL with V
End

The computing model initiates a block size (BS) of [n x n x n], and the entire sequence (Vs) is arranged in columns of BS with the size n x n, as shown below:

$$\begin{bmatrix} p_{1,1} & p_{1,k} & p_{1,n} \\ \cdots & \cdots & \cdots \\ p_{m,1} & p_{m,k} & p_{m,n} \end{bmatrix} \rightarrow \text{f(BS):[B1,B2,B3,….Bq]} \rightarrow [\text{B}]_{\ 1 \text{ x q}}$$

The vector size for storing the lexicon blocks in the dictionary is n^3 x q. The observation block size (oBS) is defined as [n x n]. The container size (Cs) for the dictionary feature lexicon is Cs [n^3, q, D], where D is the dimension as in equation (1) if D n ≤ 0.49 and as in equation (2) if D ≥ 0.49

$$D = \frac{\lfloor N(Vs)_t \rfloor}{n} \quad (1)$$

$$D = \frac{\lceil N(Vs)_t \rceil}{n} \cdots \quad (2)$$

For∀ Vs.∈ Sc, an empty vector $\vec{f}$ of the number of rows and columns as of Vs. The dimension of 'n' is created, and the pixels of ∀ Vs.∈ Sc is stored in all the nul1 matrices of $\vec{f}$. The Lexicon block dictionary for ∀ Vs.∈ Sc as a null matrix of size [n^3, q,]. The mean of the B is computed as $\vec{M}$ and further, the normalized Vector (V) is computed as in equation (3).

$$V = \frac{B_j - \vec{M}}{\left\| B_j - \vec{M} \right\|} \cdots \quad (3)$$

The dictionary for the feature lexicon gets updated with the corresponding values of the V.

C. Feature Engineering

The multidimensional array for the Lexicon of 'Vs' dictionary as $\vec{L}$(m x n x d) is an input vector for the feature engineering process. The 'Fs' A structure for storing the feature vectors ($\overrightarrow{Fv}$).

Algorithm-2: Feature Engineering

Input: $\vec{L}$
Output: $\overrightarrow{Fv}$
Start
for∀ n ∈ $\vec{L}$, compute $\overrightarrow{Lt}$(m,1, d)
initiate, X[0](m x 1)
for ∀ d ∈ $\overrightarrow{Lt}$
[val](m x d) ←$\overrightarrow{Lt}$(d)
end
check for noise (NAN)
$\vec{X}$ [0:NAN]←NAN: *(Algorithm-3)*
Define the number of clusters: k
Invoke, $f_{\textbf{Non-NLS}}$ *(): (Algorithm-4)*
{features}←NNLS($\vec{X}$,k)
Updates, $\overrightarrow{Fv}$*({features}*
End

This phase of the study incorporates an efficient feature engineering modeling to normalize the lexicon attributes $\vec{L}$(m x n x d) ∈ ′ Vs. ′ dictionary. The prime underlying motive of the feature engineering modeling in this research phase is to speed up the calculations during the execution phase of the empirical decomposition model. A matrix structure 'Fs' is further created to update the trainable extracted features for the feature vectors ($\overrightarrow{Fv}$). The algorithm finally yields a vector of computed ($\overrightarrow{Fv}$). The computation steps exhibit that it initially computes and creates a structure corresponding to the lexicon attributes of the dictionary feature set. Here, for each lexicon attribute of the dictionary feature matrix, the process computes each column corresponding to $\vec{L}$ Moreover, further computed in a decomposed form as: $\overrightarrow{Lt}$(m,1,d) ∈ $\vec{L}$(m x n x d). The matrix decomposition process makes the computational process efficient from the execution and memory "*S*" viewpoint. Here the computation takes place with one column vector from the matrix. The analytical algorithm also initializes another form of the matrix: X[0](m x 1), and for each individual *d*, the process workflow computes dominant and significant attributes of the Lexicon from $\overrightarrow{Lt}$(d) and further store it into [val](m x d). The dimensionality (∂) of computed feature elements gets reduced from $O(d^3) \rightarrow O(d^2)$. The execution workflow of this Algorithm 2 further checks for the noise elements (NAN) in the feature set and also performs correction of feature attributes by replacing the elements of $\vec{X}$ [0:NAN]←NAN by 0. The process of computing the adjusted and normalized feature matrix is shown with simplified execution steps below.

Algorithm-3: Normalization of the [val](m x d) ←$\overrightarrow{Lt}$(d)

Start

1. Create one structure: Feature Set
2. For each column of $\vec{L}$
   a. $\vec{L}$(m x n x d) →$\overrightarrow{Lt}$(m,1,d)
   b. X(m,1)=[0]
   c. X[val](m x d) ←$\overrightarrow{Lt}$(d)
   d. Check for NAN if found, replace it by 0, X [0: NAN]←NAN

End

Algorithm-3 computationally executes the normalization of the feature lexicon from the dictionary attributes, making them suitable to train the proposed model with higher efficiency and optimized computation. Here, the appropriate feature lexicons also improve the learning-based performance towards the accuracy of event detection. In this computing stage, the proposed model also incorporates another cost-effective approach of feature lexicon approximation by defining several clusters in k. Here the system invokes a functional segment $f_{NNLS}()$: to be operated on the normalized feature lexicon vector X[val] (m x d). Finally, the computed factorize multivariate lexicon blocks of learning-based features are evaluated by the functional component of the non-negative least square approach, which optimizes the non-negative factorization of the lexicon block matrix. The executable functional segment of $f_{NNLS}()$ is further discussed in the subsequent section. The computed trainable lexicon vector features are then stored in a structure called $\overrightarrow{Fv}\{features\}$ and it is updated. The raw form of Lexicon features Vector $\vec{X}$ (Fig. 1) before NNLS-based lexicon factorization is visualized as follows:

D. The NNLS Algorithm

The explicit function, sparse-"non-negative factorization" as S-NNF, effectively handles multivariate for the depreciation of massive matrix computation's computational resources. The algorithm for S-NNF takes the matrix of the feature vector $\overrightarrow{Fv}$. The dictionary features lexicon block as a mixed-signal to factorize organized as samples in the column and features in a row with clusters (k).

Algorithm-4: The NNLS Algorithm

Input:$\overrightarrow{\boldsymbol{Fv}}$, k
Output: $(\overrightarrow{\boldsymbol{Fv}})\boldsymbol{res}$
Process:
Start:
$\vec{Y}\leftarrow$ Matrix with random value
for each iteration
$\vec{A}\leftarrow$ (Moore-pseudo inverse $\vec{Y}$) x $\overrightarrow{\boldsymbol{Fv}}$
Update normalized $(\vec{A})$
$\vec{Y}\leftarrow f$(Ae,$(\overrightarrow{Fv})o$) // function for solution of least square
Initialize, a large value of Xp
$(\overrightarrow{Fv})c\leftarrow\vec{A}$ x $\vec{Y}$
$(\overrightarrow{Fv})f\leftarrow f((\overrightarrow{Fv})p-(\overrightarrow{Fv})c)$ *// function for the Frobenius norm*
Update$(\overrightarrow{Fv})p\leftarrow(\overrightarrow{Fv})c$
$(\overrightarrow{Fv})res\leftarrow\|\overrightarrow{Fv}$ - $(\overrightarrow{Fv})c\|$
end
End

The NNLS(): The $\vec{X}$ as $\overrightarrow{Fv}$contains -ve elements; therefore, directly non-negative factorization(NNF) is not applicable as in NNF, all $\vec{X}$ and its factors $\vec{W}$ and $\vec{H}$ Shall be having non-negative elements. Therefore, Semi-NNF is used. The semi-non-negative matrix factorization is a technique that learns a low-dimensional dataset representation that lends itself to a clustering interpretation.). A vector $\vec{Y}$as a matrix to hold the coefficient initially consist of random values with the number of rows is equal to the number of clusters(k) and the number of the columns as the number of columns of $\overrightarrow{Fv}$ and another vector $\vec{A}$ As a base matrix. In each iteration of computation, the Moore-pseudo inverse of the Vector $\vec{Y}$ updates the value of $\vec{A}$ after multiplying to the $\overrightarrow{Fv}$ and gets normalized $\vec{A}$. The parameter Ae and $(\overrightarrow{Fv})o$ is taken as an input argument to an algorithm for the solution of the NN-constraint least square problem (LSP) [AR]. With the value of the $\vec{A}$ based on the Ae and $(\overrightarrow{Fv})o$, this algorithm solves for the optimal value of the K in a least-square using equation (4).

$$\text{Ae} = (\overrightarrow{Fv})o \text{ x K} \tag{4}$$

In the problem of Min $\|$ Ae - $(\overrightarrow{Fv})o$ x K $\|$ such that K$\geq$ 0 for a given Ae and $(\overrightarrow{Fv})o$. A considerable value of the previous fit value $(\overrightarrow{Fv})p$and the current fit value $(\overrightarrow{Fv})c = \vec{A}$ x $\vec{Y}$ provides the Frobenius norm to give the fittest result $(\overrightarrow{Fv})f$ and the current fit value is updated as a previous fit value, and finally, the final residual $(\overrightarrow{Fv})res$ is gets updated using equation (5).

$$(\overrightarrow{Fv})res\leftarrow\|\overrightarrow{Fv} - (\overrightarrow{Fv})c\| \tag{5}$$

The updated value of the final residual as $(\overrightarrow{Fv})res$ at every iteration are computed where the proposed NNLS algorithm optimizes the factorization approach and computes the learning based-features of lexicon blocks in $(\overrightarrow{Fv})res$ after the completion of the training. The proposed system further extracts the test data (GT$\leftarrow$D$_{test}$) from the repository location (R_l). The process also checks for the GT dimension and converts it into greyscale form with the dimension of (∂). Here the study invokes an explicit function $f_t(x)$ to test the S-NNF of the least square, which is obtained as $(\overrightarrow{Fv})res$. The study further discussed the explicit functions of ground truth (GT) computation solution considering the formulated approach.

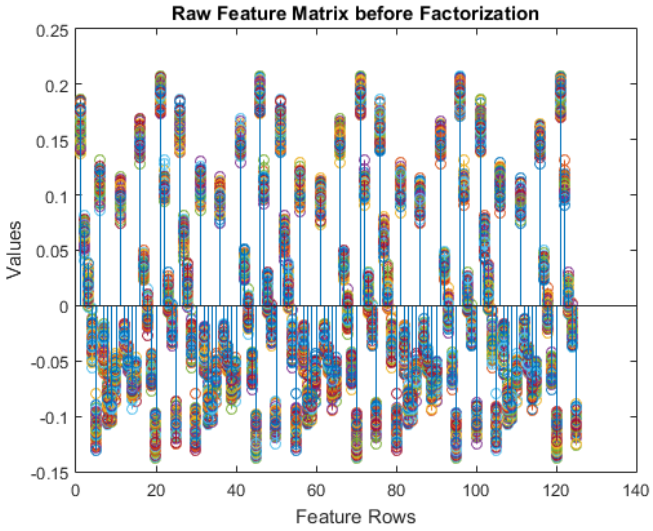


Fig. 1. Visualization of Feature Elements of the Lexicon Vector $\vec{X}$.

Explicit $f_t(x)$ to test the S-NNF of least square with GT

Input: GT, $bw \leftarrow$DFL{} , $(\overrightarrow{Fv})res, d$
Output: Updated $bw \leftarrow$DFL{}, d
Process start:

Initialize: GT, $bw \leftarrow$DFL{} , $(\overrightarrow{Fv})res, d$
$p \leftarrow R(GT)$
$C \leftarrow con(1, p)$
$if\ C \rightarrow []$
$Compute: c1 \rightarrow C(row, 1)$
$Normalize \rightarrow range(C1)$
$Compute: c2 \rightarrow C(row, 2)$
$Normalize \rightarrow range(C2)$
$update: bw \leftarrow Obj(bw)\ [func\_in: bw, c1, c2]$
$else$
$update: bw \leftarrow Z[size(GT)]$
end
$perform\ gt \leftarrow bwMO(GT)$
$bw \leftarrow bw \times gt$
$update: bw \leftarrow DFL(d)$

End

In this process, the system initially computes the GT as a binarize form of the D_{test} from the greyscale matrix. The approach here also considers the lexicon properties of DFL for the test sequence generation. Initially, the function computes the region properties from the centroid matrix corresponding to the GT. Further, the function $f_t(x)$ also performs concatenation on the centroid factor of computed region properties and stores the numerical values into a vector *C*. Further algorithmically, the computational process also assesses a conditional check on this computed Vector C. If it finds C as an empty matrix, it computes the 1st column values corresponding to *C* and stores it into another variable, *c1*. The range of *c1* is also get adjusted for ease of computation. Further, the system also computes another variable, c2, from the second column of *C* and adjusts the range of *c2*. Here the function $f_t(x)$ invokes another function to identify the objects in the binary form of an image that overlaps with the pixel attributes (pi). The function here considers bw, c1, and c2 for the execution purpose. Else the process constructs a vector of zeros for bw with the size of the matrix GT. Finally, the computational process applies morphological operations on the pixel of the binary image to identify the Obj and update the bw matrix form. In the ground truth (GT), the contiguous region is considered a connected component, sometimes known as blobs. An example of a label matrix containing the blobs is as in the following Table II:

TABLE II. LABEL MATRIX

1	1	0	2	2	0	3
1	1	0	2	2	0	3

It returns a [1 x n] vector that specifies the center of the region's area. The very first element of the centroid is the x-coordinate and the second element is the y-coordinate. The study further incorporates another functional module, $f_m(x)$, to compute the Moore-pseudo inverse $\vec{Y}$ concerning $\overrightarrow{Fv}$ as shown in the above algorithm-4 for NNLS computation and optimization procedure. This explicit function computation is discussed as follows.

Explicit $f_m(x)$ to compute Moore-pseudo inverse $\vec{Y}$

Input: $\vec{Y}$, $\overrightarrow{Fv}$, δ
Output: $\vec{A}$
Process start:

Init: $\vec{Y}$, $\overrightarrow{Fv}$, δ
Compute: [*row*, *col*] $\leftarrow$size($\vec{Y}$) matrix form
Check: func: input *arg*
If *arg* <2
If (*row* < *col*)
$\vec{A} \leftarrow \frac{(\vec{Y}' \times \vec{Y})}{\vec{Y}'}$
Else
$\vec{A} \leftarrow \frac{\vec{Y}'}{(\vec{Y}' \times \vec{Y})}$
End
If (*row* < *col*)
$\vec{A} \leftarrow \frac{1}{\delta} \times \sum \mathrm{I}(\mathrm{col}), \vec{Y}'/\vec{Y}' \times \vec{Y}$
Else
$\vec{A} \leftarrow \frac{\vec{Y}'}{\frac{1}{\delta} \times \sum \mathrm{I}(\mathrm{col}), \vec{Y}' \times \vec{Y}}$
End

The above function for Moore-pseudo inverse computation is analytically modeled in such a way that it takes $\vec{Y}$, $\overrightarrow{Fv}$ δ as inputs here δ refers to a scalar parameter that produces a stable outcome during the computation, and its value should be tremendous. Further, the process computes the size of the matrix form of $\vec{Y}$ and according to the conditional check, it executes the numerical computation of pseudo-inverse and stores it into $\vec{A}$ in the inverse computed form. The system was further subjected to compute another functional module $f_{norm}(x)$ for the Frobenius norm computation considering $(\overrightarrow{Fv})c$ as input which is the normalized form of the computed Vector $\vec{A}$. The following eq can obtain the computation of the Frobenius norm. (1).

$$\overrightarrow{Fv} \leftarrow \sqrt{\sum \sum (\overrightarrow{Fv}) c^2} \tag{6}$$

The system computes the Frobenius norm during the execution of the proposed NNLS algorithm to strengthen the feature computation and training procedure. It also applies another functional strategy of fast combinatorial to deal with the optimization problem of least square execution mode in NNLS. The function $\boldsymbol{f_O(x)}$ here adjusts the least square problem in $\vec{A}$e and computes a solution matrix k in the form of $\vec{Y}$ During the execution mode of NNLS. For an optimal matrix of $\vec{Y}$ computation, the function $\boldsymbol{f_O(x)}$ takes the input parameters $\vec{A}$e and defined coefficient matrix from ($\overrightarrow{Fv}$)o . The optimization problem for the function $\boldsymbol{f_O(x)}$ is formulated as:

$$Min\ f_O(x) \rightarrow ||\vec{A}\mathrm{e}-, (\overrightarrow{Fv})o \times k \tag{7}$$

$$subjected\ k \geq 0\ for\ given\ \vec{A}\mathrm{e}(\overrightarrow{Fv})o,$$

The proposed solution in $\vec{Y}$ in the form of k obtained from minimizing the function $f_O(x)$, the study here also checked whether the proposed NNLS algorithm converged properly toward the optimality of $\vec{Y}$. The formulated learning model based on NNLS also computes a scalar *K* for linear kernel

computation. It constructs a function $f_K(x)$ to compute K, that is, kernel vector concerning column vectors of $c1, c2$. The function $f_K(x)$ is explicitly designed as follows:

Explicit $f_K(x)$ to compute linear kernel matrix

Input: $c1, c2$
Output: K
Process start:
Init: $c1, c2$
If size (c1) >1
$c1 \leftarrow m(c1,3)$
$c2 \leftarrow m(c2,3)$
$c1 \leftarrow c1'$
$c2 \leftarrow c2'$
End
$K \leftarrow c1' \times c2$
End

The function here computes a linear kernel matrix to ease the identification of unusual objects during the tracking phase. The formulated approach computes two distinct column vectors and performs normalization of column vectors with a function $m(x)$. Finally, the product of $c1'c2$ are stored into K as a linear kernel matrix. The study constructs another function, $f_{KM}(x)$, to compute the kernel matrix for different kernel functions. Finally, for the computation of the kernel function, the $f_{KM}(x)$ evaluates the kernel function to retain the value of K. The next segment of the study discusses the experimental outcome obtained for different random test instances during the model evaluation and validation phase considering test sequence exploration concerning GT and the context of tracking unusual events.

V. Result Analysis

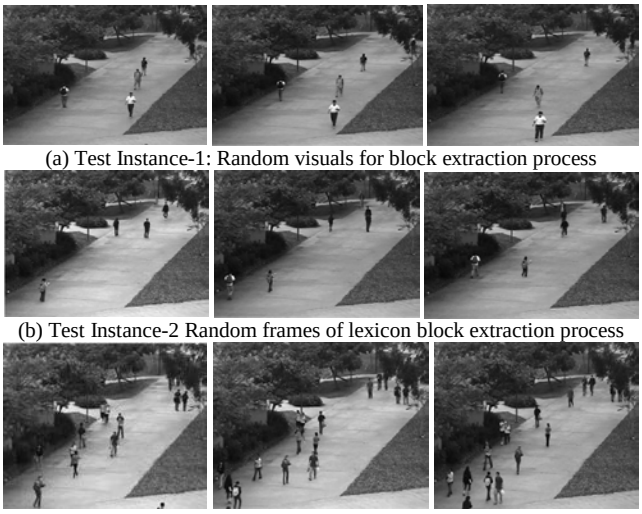
The study performs an extensive numerical analysis by investigating the outcome of a block-shift feature algorithm-based learning model for tracking unusual object movement from a video sequence. Every functional module associated with the formulated approach is evaluated with numerical modeling and a systematic execution flow. The optimized version of the proposed NNLS algorithm for S-NNF converges towards a fixed point. It helps in efficient and faster tracking of the motion patterns associated with the non-pedestrian entities. The study refers to the dataset in [38] for the block-shift feature-based framework's entire design and numerical analysis phase.

A. Analysis of the Dataset for the Experiment

The dataset contains a set of training videos and testing videos as GT for validation, and also it consists of cumulatively 5000 frame sequences for videos. Here, each video data was captured through a camera installed on the roadside to record pedestrians' feeds and the non-pedestrian pattern of movement, which is also considered an unusual movement in this study. Here, during the numerical modeling and computation of scene sequences and dictionary lexicon block extraction, it is realized that each of the moving scene sequences is composed of a set of people walking on streets and moving in two directions. Among the crowd of people, the prime role of the formulated NNLS-based approach is to track significant unusual events in the form of the non-pedestrian pattern of movement and dynamic movement of the pedestrian. The dataset of video sequences also consists of metadata annotation and a GT set of sequences in test data form. Here the annotation form of metadata indicates binary flags for each video scene. The prime intention here is to objectify the significant events to be tracked. Here unusual event tracking belongs to a cart between pedestrians, a wheelchair rolling in sideways, skaters moving in between the walking way, and a biker). The numerical modeling-based framework assesses different training and test instances to validate the performance of an NNLS-based formulated unique event tracking algorithm. It also assesses the algorithm's capability to differentiate the significant events (i.e., unusual events) from normal circumstances (i.e., pedestrians walking on the road). The visualization of the outcome for the video sequence exploration phase partially has already been shown in Fig. 2 above for training and testing sequences of random video scenes. Fig. 2 shows the unexpected visual outcome for feature block extraction during the extended video sequence exploration phase.

Fig. 2 shows the random visuals of scenes involving pedestrians on the road for a different set of training sequences in the event tracking dataset. The data of random visuals are generated during the execution of algorithm-1 for the lexicon block extraction process and feature selection modeling processes. Every algorithm modeling is analytically simplified so that the formulated learning model based on the NNLS algorithm does not encounter convergence problems during the feature evaluation process during run-time. The dataset computed post feature extraction process subjected the NNLS optimization procedure to converge towards optimal DFL with properly trained classes. During the training procedure, the visuals of unusual events are learned, tracked with the appropriate feature modeling, and reposited and indexed with the composition of a matrix set in the subsequent computation process. The proposed system of NNLS-based computation results in a single form of matrix composition with the outcome of training, making the evaluation and validation process computationally faster during run-time. The visuals of the unusual tracing of objects using the block shift feature algorithm are as follows in Fig. 3.

Fig. 3 shows the tracking visuals of the unusual movement of an object using the proposed NNLS algorithm. The validation phase is carried out concerning the computed GT, generated with the explicit function ft(x). To measure the effectiveness of the proposed tracking algorithm, the performance parameters are compared with a highly cited related study by Fang et al. [39], in which the learning model is designed based on deep learning. The comparative outcome in figure xx shows that the formulated approach attains better unusual event tracking accuracy concerning the performance parameters such as recall factor, precision score, specificity score, F1-score, and algorithm execution time. The interpretation of figure xx clearly shows that the formulated NNLS attain a significantly lesser processing time for execution, approximately 0.266621 sec.

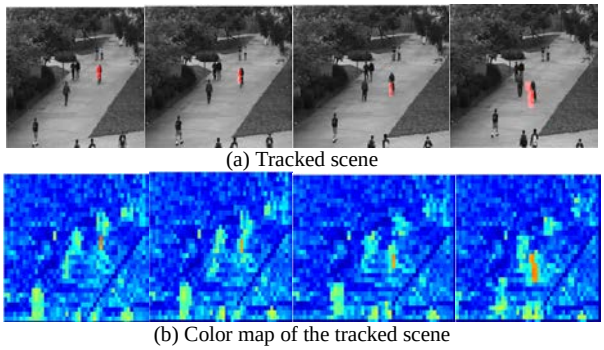


(a) Test Instance-1: Random visuals for block extraction process

(b) Test Instance-2 Random frames of lexicon block extraction process

(c) Test Instance-3: Random visuals of feature lexicon block extraction process

Fig. 2. Random Visuals of the Video Scenes from the Feature Engineering Feature Lexicon Block Extraction Process.



(a) Tracked scene

(b) Color map of the tracked scene

Fig. 3. Tracking Visuals for Training Data-1 in different Frame Instances with the Colormap.

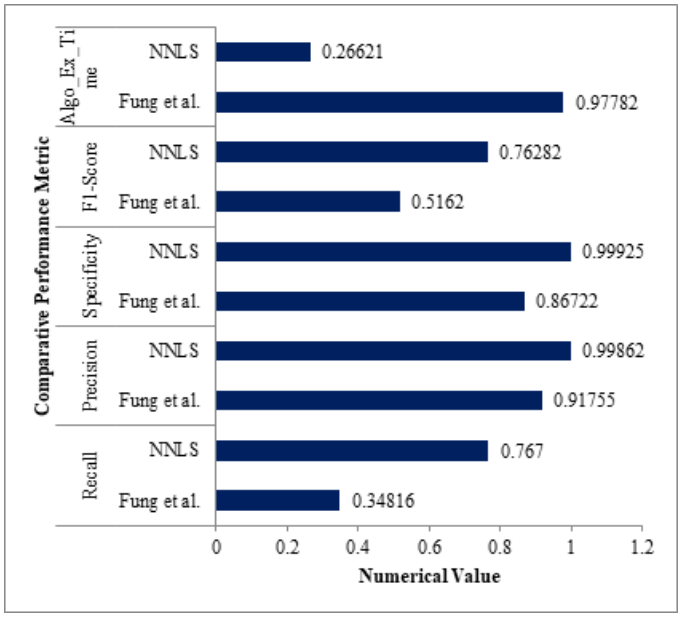


Fig. 4. Comparable Outcomes for the Effectiveness Measure in Comparison with Fang et al. [39].

In contrast, the deep learning-based model in Fang et al. [39] takes a comparatively higher computing time execution of approximately 0.97782 sec. The numerical assessment for the comparative analysis is performed in a similar test environment. The relative performance outcome is shown in the following Fig. 4.

The prime reason behind the effectiveness of the outcome corresponding to the formulated approach NNLS is that it has a lower dependency on the computational resources, unlike the deep learning models. The deep learning-based models have a more considerable dependency on the quantity of the training data set, which can attain accuracy but compromise the computational performance. However, the formulated NNLS-based tracking approach intelligently extracts features and poses lower dependency on the training images with faster execution in run time, making it suitable for real-time video tracking applications.

VI. Conclusion

This paper introduces an efficient scheme of unusual movement tracking from video sequences considering a novel block-shift feature and NNLS algorithm. The study finds a standard dataset of different video sets where the scenes comprise crowd and pedestrian movements. The underlying motive behind this research study is to track the unusual dynamics associated with a mobile object that differs from the pedestrian movement pattern. The proposed research introduces an efficient feature extraction algorithm with a block-wise feature lexicon. It optimizes the NNLS algorithm to optimize the computing and training performance of the learning model. The extensive performance outcome shows that the formulated NNLS-based approach involves training for the learning model, which doesn't include many dependencies on the computational resources and the video feeds, unlike other traditional learning models. Most conventional training models demand a comparatively larger size of trained data to accomplish better tracking accuracy. The NNLS design is optimized so that it can efficiently train with even low or medium training data and performs with higher accuracy in identifying unusual events. The formulated approach's faster and timelier execution makes it highly applicable in a practical environment.

References

[1] R. Fan, F-L Zhang, M.Zhang, "Robust tracking-by-detection using a selection and completion mechanism," Springer, Computational Visual Media, Vol. 3, No. 3, September 2017, 285–294, DOI 10.1007/s41095-017-0083-7.

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

[3] Y. Yuan, Y. Lu, and Q. Wang, "Tracking as a Whole: Multi-Target Tracking by Modeling Group Behavior With Sequential Detection," in IEEE Transactions on Intelligent Transportation Systems, vol. 18, no. 12, pp. 3339-3349, Dec. 2017, doi: 10.1109/TITS.2017.2686871.

[4] C. Liu, H. Chen, K. Lo, C. Wang and J. Chuang, "Accelerating Vanishing Point-Based Line Sampling Scheme for Real-Time People Localization," in IEEE Transactions on Circuits and Systems for Video Technology, vol. 27, no. 3, pp. 409-420, March 2017, doi: 10.1109/TCSVT.2017.2649019.

[5] D. De Gregorio, R. Zanella, G. Palli, S. Pirozzi and C. Melchiorri, "Integration of Robotic Vision and Tactile Sensing for Wire-Terminal Insertion Tasks," in IEEE Transactions on Automation Science and Engineering, vol. 16, no. 2, pp. 585-598, April 2019, doi: 10.1109/TASE.2018.2847222.

[6] C. Cuevas, R. Martínez, D. Berjón, and N. García, "Detection of Stationary Foreground Objects Using Multiple Nonparametric Background-Foreground Models on a Finite State Machine," in IEEE Transactions on Image Processing, vol. 26, no. 3, pp. 1127-1142, March 2017, doi: 10.1109/TIP.2016.2642779.

[7] T. Huynh-The, C. Hua, N. A. Tu and D. Kim, "Locally Statistical Dual-Mode Background Subtraction Approach," in IEEE Access, vol. 7, pp. 9769-9782, 2019, doi: 10.1109/ACCESS.2019.2891084.

[8] C. Liang and C. Juang, "Moving Object Classification Using a Combination of Static Appearance Features and Spatial and Temporal Entropy Values of Optical Flows," in IEEE Transactions on Intelligent Transportation Systems, vol. 16, no. 6, pp. 3453-3464, Dec. 2015, doi: 10.1109/TITS.2015.2459917.

[9] H. Dou, D. Ming, Z. Yang, Z. Pan, Y. Li, and J. Tian, "Object-Based Visual Saliency via Laplacian Regularized Kernel Regression," in IEEE Transactions on Multimedia, vol. 19, no. 8, pp. 1718-1729, Aug. 2017, doi: 10.1109/TMM.2017.2689327.

[10] D. P. Dogra, A. K. Majumdar, S. Sural, J. Mukherjee, S. Mukherjee, and A. Singh, "Toward Automating Hammersmith Pulled-To-Sit Examination of Infants Using Feature Point-Based Video Object Tracking," in IEEE Transactions on Neural Systems and Rehabilitation Engineering, vol. 20, no. 1, pp. 38-47, Jan. 2012, doi: 10.1109/TNSRE.2011.2172223.

[11] J. Dunik, O. Straka, M. Simandl, and E. Blasch, "Random-point-based filters: analysis and comparison in target tracking," in IEEE Transactions on Aerospace and Electronic Systems, vol. 51, no. 2, pp. 1403-1421, April 2015, doi: 10.1109/TAES.2014.130136.

[12] M. Gupta, S. Kumar, L. Behera, and V. K. Subramanian, "A Novel Vision-Based Tracking Algorithm for a Human-Following Mobile Robot," in IEEE Transactions on Systems, Man, and Cybernetics: Systems, vol. 47, no. 7, pp. 1415-1427, July 2017, doi: 10.1109/TSMC.2016.2616343.

[13] T. Zhang and S. Fei, "Improved particle filter for object tracking," 2011 Chinese Control and Decision Conference (CCDC), Mianyang, 2011, pp. 3586-3590, doi: 10.1109/CCDC.2011.5968843.

[14] Jong-Min Jeong, Tae-Sung Yoon, and Jin-Bae Park, "Kalman filter-based multiple objects detection-tracking algorithm robust to occlusion," 2014 Proceedings of the SICE Annual Conference (SICE), Sapporo, 2014, pp. 941-946, doi: 10.1109/SICE.2014.6935235.

[15] N. Widynski and M. Mignotte, "A MultiScale Particle Filter Framework for Contour Detection," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 36, no. 10, pp. 1922-1935, Oct. 2014, doi: 10.1109/TPAMI.2014.2307856.

[16] Shen, Chunhua& Kim, Junae& Wang, Hanzi. (2010). Generalized Kernel-Based Visual Tracking. Circuits and Systems for Video Technology, IEEE Transactions on. 20. 119 - 130. 10.1109/TCSVT.2009.2031393.

[17] Chen, Zezhi&Husz, Zsolt& Wallace, Iain & Wallace, Andrew. (2007). Video Object Tracking Based on a Chamfer Distance Transform. Proceedings - International Conference on Image Processing, ICIP. 3. 357-360. 10.1109/ICIP.2007.4379320.

[18] Y. Wang and J. Zhang, "Application of SVM in Object Tracking Based on Laplacian Kernel Function," 2016 8th International Conference on Intelligent Human-Machine Systems and Cybernetics (IHMSC), Hangzhou, 2016, pp. 557-561, doi: 10.1109/IHMSC.2016.121.

[19] K. Sun, W. Tao and Y. Qian, "Guide to Match: Multi-Layer Feature Matching With a Hybrid Gaussian Mixture Model," in IEEE Transactions on Multimedia, vol. 22, no. 9, pp. 2246-2261, Sept. 2020, doi: 10.1109/TMM.2019.2957984.

[20] Q. Lin et al., "Robust Stereo-Match Algorithm for Infrared Markers in Image-Guided Optical Tracking System," in IEEE Access, vol. 6, pp. 52421-52433, 2018, doi: 10.1109/ACCESS.2018.2869433.

[21] Q. Zhu, H. Xiong, and X. Jiang, "Shape-oriented segmentation with graph matching corroboration for silhouette tracking," 2012 Visual Communications and Image Processing, San Diego, CA, 2012, pp. 1-6, doi: 10.1109/VCIP.2012.6410762.

[22] Z. Boukhers, K. Shirahama, and M. Grzegorzek, "Example-Based 3D Trajectory Extraction of Objects From 2D Videos," in IEEE Transactions on Circuits and Systems for Video Technology, vol. 28, no. 9, pp. 2246-2260, Sept. 2018, doi: 10.1109/TCSVT.2017.2727963.

[23] S. Damotharasamy, "Approach to model human appearance based on sparse representation for human tracking in surveillance," in IET Image Processing, vol. 14, no. 11, pp. 2383-2394, 18 9 2020, doi: 10.1049/iet-ipr.2018.5961.

[24] L. Kong, D. Huang, J. Qin, and Y. Wang, "A Joint Framework for Athlete Tracking and Action Recognition in Sports Videos," in IEEE Transactions on Circuits and Systems for Video Technology, vol. 30, no. 2, pp. 532-548, Feb. 2020, doi: 10.1109/TCSVT.2019.2893318.

[25] Y. Lee, Z. Tang and J. Hwang, "Online-Learning-Based Human Tracking Across Non-Overlapping Cameras," in IEEE Transactions on Circuits and Systems for Video Technology, vol. 28, no. 10, pp. 2870-2883, Oct. 2018, doi: 10.1109/TCSVT.2017.2707399.

[26] C. Liu, R. Yao, S. H. Rezatofighi, I. Reid and Q. Shi, "Model-Free Tracker for Multiple Objects Using Joint Appearance and Motion Inference," in IEEE Transactions on Image Processing, vol. 29, pp. 277-288, 2020, doi: 10.1109/TIP.2019.2928123.

[27] A. Ratre and V. Pankajakshan, "Tucker tensor decomposition-based tracking and Gaussian mixture model for anomaly localization and detection in surveillance videos," in IET Computer Vision, vol. 12, no. 6, pp. 933-940, 9 2018, doi: 10.1049/iet-cvi.2017.0469.

[28] S. Tian, X. Yin, Y. Su, and H. Hao, "A Unified Framework for Tracking Based Text Detection and Recognition from Web Videos," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 40, no. 3, pp. 542-554, 1 March 2018, doi: 10.1109/TPAMI.2017.2692763.

[29] C. Yang et al., "Tracking Based Multi-Orientation Scene Text Detection: A Unified Framework With Dynamic Programming," in IEEE Transactions on Image Processing, vol. 26, no. 7, pp. 3235-3248, July 2017, doi: 10.1109/TIP.2017.2695104.

[30] S. C. Wong, V. Stamatescu, A. Gatt, D. Kearney, I. Lee, and M. D. McDonnell, "Track Everything: Limiting Prior Knowledge in Online Multi-Object Recognition," in IEEE Transactions on Image Processing, vol. 26, no. 10, pp. 4669-4683, Oct. 2017, doi: 10.1109/TIP.2017.2696744.

[31] X. Cheng, Y. Gu, B. Chen, Y. Zhang, and J. Shi, "Weighted Multiple Instance-Based Deep Correlation Filter for Video Tracking Processing," in IEEE Access, vol. 7, pp. 161220-161230, 2019, doi: 10.1109/ACCESS.2019.2951600.

[32] A. Khan et al., "Forensic Video Analysis: Passive Tracking System for Automated Person of Interest (POI) Localization," in IEEE Access, vol. 6, pp. 43392-43403, 2018, doi: 10.1109/ACCESS.2018.2856936.

[33] X. Lan, M. Ye, R. Shao, B. Zhong, P. C. Yuen, and H. Zhou, "Learning Modality-Consistency Feature Templates: A Robust RGB-Infrared Tracking System," in IEEE Transactions on Industrial Electronics, vol. 66, no. 12, pp. 9887-9897, Dec. 2019, doi: 10.1109/TIE.2019.2898618.

[34] X. Liu, Y. Xu, L. Zhu, and Y. Mu, "A Stochastic Attribute Grammar for Robust Cross-View Human Tracking," in IEEE Transactions on Circuits and Systems for Video Technology, vol. 28, no. 10, pp. 2884-2895, Oct. 2018, doi: 10.1109/TCSVT.2017.2781738.

[35] Van Benthem MH, Keenan MR. A fast algorithm for the solution of large-scale non-negativity-constrained least squares problems. Journal of Chemometrics: A Journal of the Chemometrics Society. 2004 Oct;18(10):441-50.

[36] Karanam Sunil Kumar and N P Kavya, "Compact Scrutiny of Current Video Tracking System and its Associated Standard Approaches" International Journal of Advanced Computer Science and Applications (IJACSA), 11(12) 2020. http://dx.doi.org/10.14569/IJACSA.2020.0111249.

[37] Kumar, K.S. and Kavya, N.P., 2021, April. Novel Approach of Video Tracking System Using Learning-Based Mechanism over Crowded Environment. In Computer Science On-line Conference (pp. 67-76). Springer, Cham.

[38] http://www.svcl.ucsd.edu/projects/anomaly/dataset.html.

[39] Z. Fang & F. Fei, & Y. Fang, & C. Lee, & N. Xiong, & L. Shu, & S. Chen, "Abnormal event detection in crowded scenes based on deep learning", Springer Journal of Multimedia Tool Application, 2016, DOI 10.1007/s11042-016-3316-3.