Communications on Applied Nonlinear Analysis
ISSN: 1074-133X
Vol 32 No. 9s (2025)

The NovelHAP Algorithm for Human Crime Activity Prediction in Video Frames using Kalman Filter and Template Matching

Y. V. K. Durga Bhavani [1*], V. B. Pagi [2]

[1]Department of CSE, Basaveshwar Engineering College Research Center, Bagalkote, Affiliated to Visvesvaraya Technological University, Belagavi-590 018, Karnataka, India

[*]***Corresponding Author Email:*** *bhavanivani35@gmail.com* ***-ORCID****:0000-0003-2435-1094*

[2]Department of CSE, Basaveshwar Engineering College Research Center, Bagalkote, Affiliated to Visvesvaraya Technological University, Belagavi – 590 018, Karnataka, India

Email*: veereshpagi@gmail.com* ***- ORCID****: 0000-0001-9555-0707*

Article History:

Received: 12-01-2025

Revised: 15-02-2025

Accepted: 01-03-2025

Abstract:
Human Activity Prediction (HAP) interprets human motion and future activity. Surveillance systems must use HAP techniques to deter crimes and dangerous behavior. The primary objective of this study is to forecast human criminal acts in video frames, encompassing shooting, fighting, and kicking. This study introduces the NovelHAP algorithm, built upon the Kalman Filter Algorithm (KFA), MediaPipe (MPP) framework, and Template-Matching Statistical Approaches (TMSA). KFA is used to predict a video frame that contains a very near state (or position) of one human Bounding Box (BB) and its centroid position compared to another in the video frames. When the two humans are very close to each other, the activity of the human can be predicted. The predicted frames have been considered as input to the MPP Framework to find the landmarks and posture of the human. This study uses the values of key landmarks Left_Wrist, Right_Wrist, Left_Hip, Right_Hip, Left_Ankle, and Right_Ankle of human postures in video frames. The Euclidean distances from D_1 to D_{10} among the landmark vectors of video frame postures (training vectors) and template vectors (test vectors) are used as the input for TMSA. TMSA measures the similarity between training and test vectors to predict and classify human crime activities. The empirical analysis revealed that the accuracies for HAP on the UCF-101 dataset are 95% for small data samples, and 96.01% for large data samples, respectively.

Keywords: Kalman Filter, Media Pipe, Prediction, Template Matching.

1. Introduction

In the fields of action classification and video processing, Human Activity Prediction (HAP) is an essential and difficult task. Several researchers have taken an interest in HAP, which combines computer vision and artificial intelligence (AI) with numerical analysis. HAP is important in various fields, including assistive robotics, smart homes, human-computer interactions, security, transport, education, and crime scene analysis [2], [7]. For several reasons, such as differences in perspective and distance from the camera, the complexity of the background, and the range of speeds, the problem remains sensitive even with the notable progress in HAP learned from video sequences.

Crime is defined as an intentional or accidental act that can result in property damage or loss, psychological or physical harm, and, depending on the seriousness of the offense, punishment from the state or another authority [6]. Agencies are being forced to create effective strategies for taking

preventive action because of the startling rate at which the quantity and types of criminal activity are growing. Traditional crime-solving techniques are slow and inefficient, making them useless in the current climate of rapidly increasing crime. Therefore, police personnel can help prevent crimes if we can devise methods to forecast crime before it happens. The majority of video-based HAP systems use previous activity video frames to categorize and forecast future activities. Crime prediction is crucial since it may save a victim's life, prevent enduring suffering, protect private property, and forecast potential terrorist activities [1].

In the current scenario of rapidly increasing crime, traditional crime-solving techniques are unable to deliver results, being slow-paced and less efficient. Thus, if we can come up with ways to predict crime at a more accurate and faster rate, before it occurs, or come up with a "machine" that can assist police officers, it would lift the burden of police and help in preventing crimes. Human crime activity predictive analysis enables law enforcers to take action before a crime occurs. HAP is still a challenging topic that has not been fully solved, because of several acquisition platform limitations, including the dynamic and complicated background, point-of-view variation, camera height, and human appearance, etc.

[16] proposed an architecture for video activity prediction that combines Saliency-aware Motion Enhancement (SME) with a Weighted Long Short-Term Memory Network (WLSTM). SMEs effectively reduce spatial and temporal redundancy interference by suppressing background noise in video frames. After evaluation, the framework decreases the temporal redundancy and achieves the highest prediction results of 98.3%, 95.1%, and 78.1% on the UT-Interaction dataset sets 1 and 2 and the sub-JHMDB dataset, respectively. [17] explains a special HAP system using Inertial Measurement Unit (IMU) anticipated activity data. Used a Bi- Long short-term memory (LSTM) deep learning classifier for predicting future activities and a deep learning forecaster for IMU activity signals comprising the HAP system. Two triaxial IMU sensors were used to test the HAP system for five daily tasks. For the five activities, the average correlation between the projected and observed signals is 91.6%. The proposed HAP system predicts future actions with an average accuracy of 97.96%. [8] introduced a multi-network fusion strategy based on a feature-based method for identifying violence or the lack of violence in real-world surveillance. Initially, a convolutional neural network (ConvNet) retrieves the spatial data. LiteFlowNet acquired the pyramidal convolutional features from two sequential frames. A deep-gated recurrent unit (GRU) independently receives the outputs from both networks (ConvNet and LiteFlowNet). Data from each GRU is combined and subsequently transferred to the dense layer for the final decision-making process. An ablation study is undertaken on three real-world surveillance datasets to assess the efficacy of approach 1) The RWF-2000 dataset obtained an accuracy of 94.50%. 2) The Surveillance Fight dataset reached an accuracy of 96.69%, and 3) the Hockey Fight dataset attained an accuracy of 98.62%.

[11] proposed Kalman filters algorithm for human tracking in urban environments online videos. The proposed methods show that the efficiency of frame-by-frame tracking depends on preserving a trace of past positions. The tracking system's efficiency is increased by including a prediction of future positions, which enables it to track people in real-world situations including sudden changes in trajectory, individual crossings, and disappearances behind obstacles. [2] describes a novel method for predicting human activity from videos using the Convolutional Neural Network (CNN) in conjunction with a new human skeletal architecture for feature representation and the Recurrent Neural Network

(RNN) for activity categorization. Kalman Filter can track a moving person in a video stream and determine the bounding box from a centroid location. The RGB-D sensor dataset CAD-60 is used to assess the effectiveness of the suggested method. The performance is demonstrated by the average error rate of 4.5% in the experimental findings.

[3] introduced the Kalman filter algorithm for tracking and detecting multi-objects in video. Adaboost classifier detected the objects and upstream tracking using the Camshift and Kalman filter algorithms. The cascade of AdaBoost classifiers improves the calculation time and the quality of tracking one or more objects in a sequence of images or videos. [12] explains the Kalman filter and integrates with the neural network technique. It comprises a succinct explanation of the traditional Kalman filter as well as its variants, including the cubature Kalman filter (CKF), unscented Kalman filter (UKF), and extended Kalman filter (EKF). The hybrid model performs better in terms of accuracy and resilience. [13] describes a Kalman filter strategy for predicting traffic patterns approaching an intersection based on data from linked automobiles. According to the investigation's findings, the Kalman filter performs best when the penetration rate exceeds 20%. Because the suggested technique uses the connected vehicle data right before the prediction period, it is computationally efficient and provides a real-time prediction.

[14] proposed the CNN with an Improved Tree Seed algorithm based on the Unified Deep Features optimization feature-based method for human activity classification. For the implementation (i) The KTH dataset was used to recognize human actions such as boxing, handclapping, handwaving, jogging, running, and walking with an accuracy of 99.6% (ii) The Weizmann dataset was used to identify human behaviors such as bending, leaping in the air, running, leaping forward, galloping sideways, strolling, skip hopping, waving one hand, and waving both hands. The stated accuracy is 98.1% (iii) The UT-Interaction dataset was tested to recognize human actions such as pushing, pointing, kicking, punching, handshaking, and hugging. The stated accuracy is 100% (iv) The Hollywood dataset was used to recognize human activities such as AnswerPhone, GetOutCar, Handshake, HugPerson, Kiss, SitDown, SitUp, and StandUp. The accuracy recorded 99.8%. (iv) The IXMAS dataset was employed to identify human activities such as CheckWatch, CrossArms, ScratchedHead, SitDown, GetUp, TurnAround, Walk, Wave, Puch, Kick, and PickUp. The stated accuracy is 97.5%.

[15] offered the ConvLSTM method, Long-Term Recurrent Convolutional Network (LRCN) algorithm, and CNN model based on a feature-based method to recognize baseball pitch, basketball shooting, bench press, biking, billiards shot, clean and jerk, diving, drumming, fencing, golf swing, and other human activities in the UCF50 and HMDB51 datasets, respectively. The ConvLSTM approach was applied to the UCF50 and HMDB51 datasets to recognize human activity (HA), with accuracies of 82% and 68%, respectively. To recognize HA, the LRCN method was used for the UCF50 and HMDB51 datasets, which yielded accuracies of 93.44% and 71.55%, respectively. The CNN model was applied to the UCF50 and HMDB51 datasets to recognize HA, with accuracies of 99.58% and 92.70%, respectively. The CNN model using the UCF50 dataset achieved the highest accuracy of 99.58%. [18] investigates the use of multi-task learning by simultaneously predicting the next activity and the timestamp of the next event, and examines LSTM neural networks for forecasting the immediate next occurrence. A variety of sequence prediction methods, including Distributed Generator (DG), All-k Order Markov (AKOM), Long Short-Term Memory (LSTM), and Prediction by Partial

Matching (PPM), were tested on a set of publicly accessible real-world datasets from the smart home environment sector.

[19] provides a paradigm for recognizing and predicting human activity. The framework consists of three stages: recognition after the action, recognition in progress, and activity prediction in advance via passive Radio-Frequency Identification (RFID) tags. The experimental result demonstrates good performance in both activity recognition and prediction with high scalability. [10] introduced a system for identifying human activity that uses algorithms and a skeleton-based MediaPipe method to determine posture. It utilizes classifiers including Gradient Boosting (GB), Random Forest (RF), Ridge Regression (RR), and Logistic Regression (LR) to recognize namaste, standing, bending down, and raising both hands and, types of activities with 98% accuracy. [20] adopted a novel Attention-Mechanism-based Deep Learning Feature Combination (AM-DLFC) model for HAR, and utilized CNN. The AM-DLFC model accurately classified the run, pick, and sit-up activities. The performance is assessed on the 1) Khulna University (KU)-HAR dataset with an accuracy of 97%, 2) the University of California Irvine (UCI)-HAR dataset with an accuracy of 94.25%, and 3) the WISDM dataset with an accuracy of 94.26%. [23] presented a unique fusion approach using the skeleton-based method for combining multi-view skeletal data to obtain accurate human posture data. The public dataset Functional Movement Screen (FMS) was used to train sophisticated algorithms for recognizing human posture. A new fusion technique compared to other Spatial Temporal-Graph Convolutional Network (ST-GCN) models; the 2s- two stream- Adaptive Graph Convolutional Network (AGCN) method's experimental findings yield a higher accuracy of 98.2%.

A variety of Machine Learning (ML) and Deep Learning (DL) techniques have been documented in the literature for human detection, tracking, human activity recognition (HAR), and human activity prediction (HAP) across different applications. The current approaches employ skeleton-based and feature-based techniques. Existing method implementations have not addressed the prediction of human criminal activities (HAP) in video frames when two individuals are near one another. However, various challenges remain in HAP concerning the attainment of high accuracy and minimal computational complexity amongst multiple visual complications, such as occlusion, background clutter, low video quality, illumination, and overlapping persons.

This paper introduces the NovelHAP algorithm, which combines a recursive optimal estimation filter, specifically the Kalman Filter Algorithm (KFA), a skeleton-based technique (Media Pipe), and template-matching statistical approaches (TMSA) to predict human criminal activities, such as shooting, fighting, and kicking, in video frames, without employing machine learning or deep learning methods. Deep learning models are typically resource-intensive when processing large datasets and less efficient with tiny data samples. Furthermore, machine learning models demonstrate diminished effectiveness in tackling significant visual challenges such as occlusion, poor video quality, and illumination.

The proposed strategy emphasizes the use of standard techniques like KFA and TMSA, which are task-specific approaches. The standard procedures guarantee **1) Interoperability**: the capacity to exchange data among diverse systems. **2) Compatibility:** functioning with one or multiple platforms **3) Reliability:** providing consistent and impeccable outcomes, and **4) Maintainability:** the capacity to access tools and documentation when problems occur. The approaches for the work at hand are organized hierarchically, breaking down complex tasks into smaller, manageable subtasks that define

goal-oriented activities. Methods focused on the work at hand were employed to enhance efficiency, reduce distractions, and concentrate on the present task. The proposed methodology has been experimentally shown to minimize computational costs for large datasets and is highly successful for small datasets and difficult challenges in critical video analysis.

The principal aim of this research is as follows:

i) Forecasting human criminal behavior in video frames with less computational complexity, omitting machine learning and deep learning techniques.

ii) Empirically validate the efficacy of this research in predicting human criminal behavior through a comprehensive investigation of human postures using landmarks.

iii) Exhibit the effectiveness of HAP on both small and big datasets in tackling significant video complexities while reducing false positives and negatives.

The proposed methodology was executed and evaluated using the UCF-101 public surveillance dataset of real-world crimes. It is utilized to analyze extensive unedited surveillance video collections for the prediction of human criminal conduct. The 13 real-world anomalies that are captured by the camera include abuse, arrests, arson, assault, car crashes, burglaries, explosions, fights, robberies, shootings, thieving, shoplifting, and vandalism. The surveillance video datasets take into account critical video complexity issues such as occlusion, background clutter, low video and image quality, illumination, overlapped people, both indoor and outdoor, moving people, small objects (people), a variety of poses, people wearing different clothing, people lying down on the ground, night videos, and various weather conditions.

Following the introduction and literature review, the paper's remaining portion is structured into three sections. The proposed methodology NovelHAP is inclined inside segment II. The outcome and analysis are addressed in part III. Sector IV offers the conclusion as well as recommendations for further investigation.

2. Proposed Methodology

The fundamental method of predicting crimes before they occur is called crime forecasting. To anticipate crimes before they happen, different approaches are needed. Human activity prediction is a probabilistic approach that enables early recognition of incomplete activities rather than post-factum classification of completed activities. Activity prediction algorithms are very important for surveillance systems to prevent crimes and risky actions from occurring. Surveillance video footage has multiple moving objects. It is a challenging task to track multiple moving objects and predict very near-located humans for surveillance largely due to camera noise, lighting conditions, object occlusion, and the varying orientation of different objects.

This work presented a novel methodology, NovelHAP, for predicting human criminal activity in video frames. Figure 1 depicts the block diagram of the NovelHAP proposed methodology. The proposed method has 3-step procedures:

1) **Kalman Filter Algorithm (Step-1):** Video input has been given to the Kalman Filter for multi-object tacking to predict a video frame that contains very near to each other state (or position) humans in video frames.

2) **MediaPipe Framework (Step-2):** The MediaPipe Pose (MPP) landmark detector feature extraction technique is used to detect landmarks and postures of the human in both predicted video frames (training data) and different activities template images (test data). Calculated Euclidean

distances (D_1 to D_{10}) between different combinations of key landmarks (Left_Wrist, Right_Wrist, Left_Hip, Right_Hip, Left_Ankle, Right_Ankle) of both predicted video frame human (train data) and template humans (test data).

3) **Template Matching and Statistical Approaches (TMSA) (Step-3):** Step-2 calculated D_1 to D_{10} values of both train and test data has been used to find the similarity. Cross-correlation (CC), Normalized Cross Correlation (NCC), and Sum of Absolute Difference (SAD) methods measured the similarity to predict a single human crime activity in video frames.

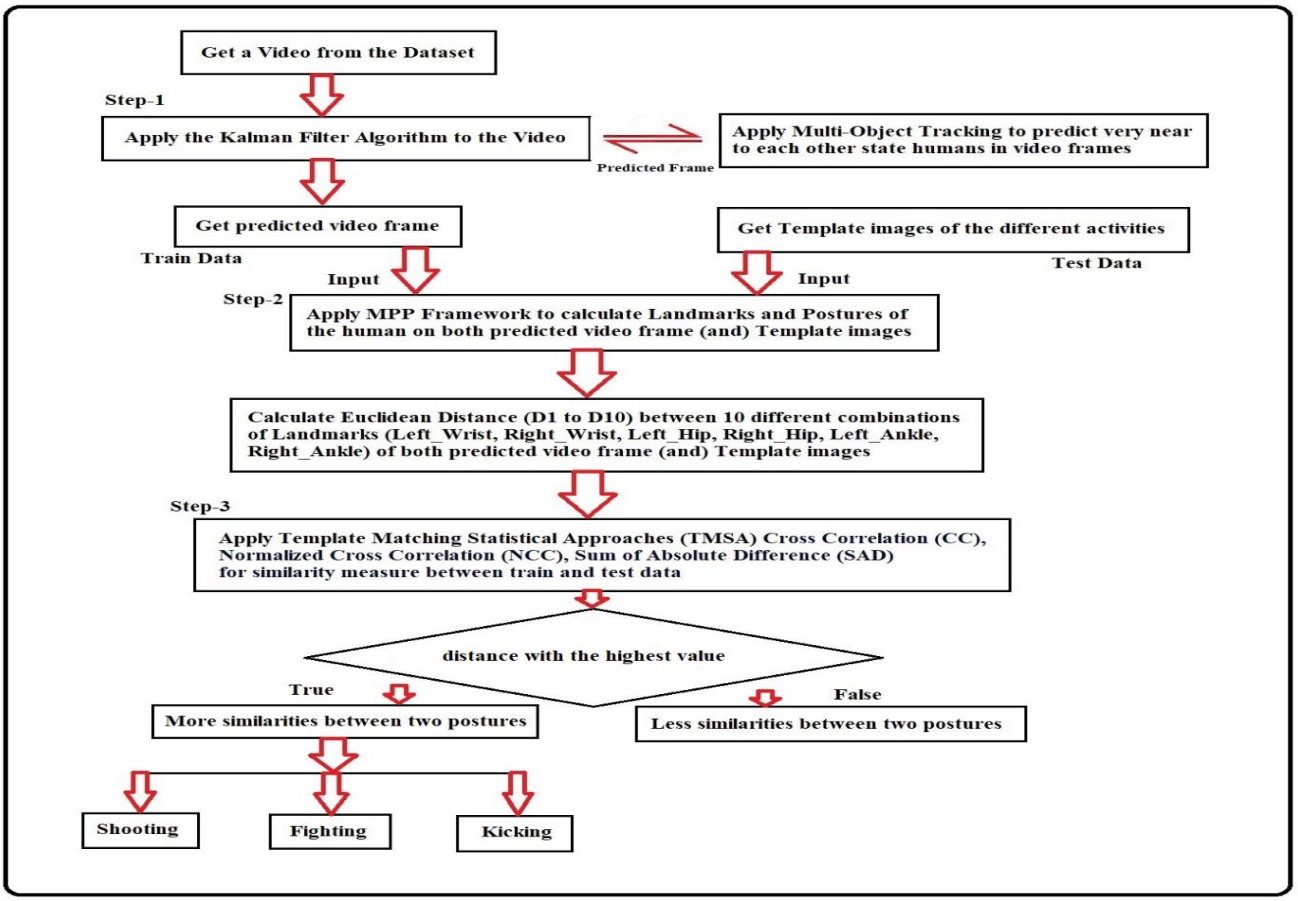

Figure 1. Block diagram of NovelHAP Proposed Methodology

2.1. Step-1: Kalman Filter Algorithm-KFA

The Kalman filter is an algorithm that assesses a system's state by integrating a sequence of measurements collected over time, accommodating statistical noise. It provides a future state prediction based on the previous estimation. The Kalman filter has been acknowledged as the ideal choice for several applications, including robotics, image systems, surveillance systems, etc, in computer vision, for instance, tracking objects [5] (mono-objects and multi-objects), prediction, and correction tasks. The Kalman filter is delineated in three stages as shown in Figures 2(a) and (b): Prediction, estimation (correction), and updating. Predicting the parameters of the tracking and detection objects is the first step. Prediction parameters are estimated and corrected in the second step. The third step is to update predictions based on errors [2], [3],[4].

The Kalman filter is a predictive correction filter. This filter examines an object as it moves during the tracking and detection process, gathering information on the object's status at that specific instant. KF uses the gathered information to predict the object's position in the subsequent frame. The Kalman Filter accepts a measurement vector comprising the object's position in x, and y coordinates, width, and height. It subsequently uses internal parameters (position, velocity, and acceleration in the x and y axes, together with height and width) to predict and assess the variables. The result is an approximation of the measurement. The Kalman filter can be employed to estimate the motion states of a moving object [3].

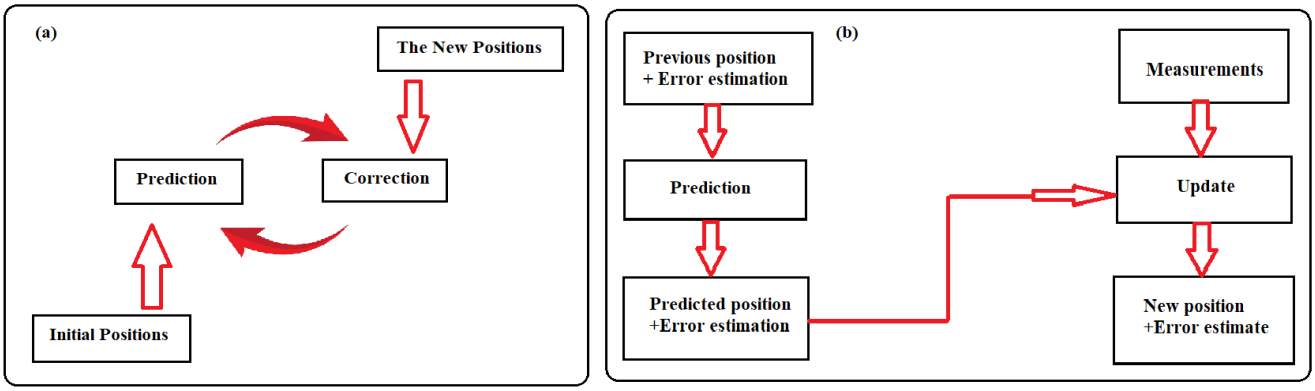

Figure 2. (a) The data flow diagram for a Kalman filter (b) Internal process of Kalman filter

This study uses the Kalman Filter to develop an object motion model that utilizes current object data to predict its position, minimizing the search area and duration for moving objects, and facilitating rapid tracking. The analysis of many objects in a video sequence is divided into three steps. The initial phase entails identifying dynamic objects. Subsequently, the phase of tracking the objects from one image to the next occurs, followed by the assessment of the trajectories to identify their activities.

2.1.1. Key aspects of Kalman Filter

The Kalman filter (KF) is a mathematical estimator utilized for predicting and correcting the states of various linear systems. It is both pragmatic and philosophically attractive; specifically, the optimal state is achieved recursively with minimal variance error. However, a precise model is necessary. The Kalman filter possesses numerous critical elements for estimating challenges in Computer Vision using Python. These components encompass: **A) State Space Model:** The KF represents the system through a series of mathematical equations that delineate the system's state, measurements, and inherent noise **B) Prediction Step:** The KF forecasts the subsequent state of the system utilizing the prior state and the mathematical model of the system **C) Correction Step:** The KF corrects the predicted state by incorporating the measurements acquired from the system.

In Kalman filtering analyses a tracking system where x_k represents the magnitude of the state vector (or location and velocity) defined in Eq. (1), which represents the object's dynamic behavior, and subscript k signifies discrete time. The objective is to estimate x_k from the measurement z_k specified in Eq. (3). The following is a four-phase mathematical description of the Kalman filter which is intended to understand the key aspects of the KF.

A) State Space Model (2 phases)

1. *Process equation:*

$$x_k = \boldsymbol{A}x_{k-1} + w_{k-1} \quad (1)$$

x_k: State at time k-1 to k
A: Transition matrix
w_{k-1} : Gaussian process noise N(.) with $p(w)$
$p(w)$: Normal probability distribution defined in Eq. (2)

$$P(w) \sim \mathrm{N}(0, \mathrm{Q}) \qquad (2)$$

P: The current state Covariance matrix of x_k.
Q: Process Uncertainty / Noise.

2. *Measurement equation:*

$$z_k = Hx_k + v_k \qquad (3)$$

Where *H:* Measurement matrix or function of x_k
z_k *:* Observations made at times *k-1* to *k,* respectively
v_k*:* *N*(.) with $p(v)$ is the Gaussian measurement noise
$p(v)$: Normal probability distribution defined in Eq. (4)

$$p(v) \sim \mathrm{N}(0, \mathrm{R}) \quad (4)$$

R: Measurement Uncertainty/noise.

B) Prediction Step

3. *Time update equations:*

Eqs. (1) and (3) provide a linear model for time k. Because x_k is not directly observed, the knowledge obtained from the measured z_k is utilized to revise the unknown state x_k. Eq. (5) is used to estimate the state $\hat{x}_k^-$ (state prediction) and calculate the covariance error $\hat{x}_k$ for the next time step k.

$$\hat{x}_k^- = A\hat{x}_{k-1} + w_k \qquad (5)$$
$$P_k^- = \mathrm{A}P_{k-1}A^T + Q \quad (6)$$

P_k^-: Error covariance prediction is calculated by using Eq. (6). Covariance measures how much movement in one variable predicts movement in the associated variables. Covariance focuses on direction, not strength.

C) Correction Step

4. *Measurement update equations:*

The system feedback and measurement update equations are represented by Equations (7), (8), and (9), respectively. The goal is to estimate $\hat{x}_k$, a linear combination of the new measurement z_k and the apriori estimate. The equations are shown below:

$$K_k = P_k^- H^T \, (\mathrm{H}P_k^- \, \mathrm{HT} \, + \, R)^{-1} \qquad (7)$$
$$\hat{x}_k = \hat{x}_k^- + K_k \, (z_k - \mathrm{H}\, \hat{x}_k^-) \qquad (8)$$
$$P_k = (1 - K_k \mathrm{H}) P_k^- \qquad (9)$$

K_k *:* Kalman gain which is computed by Eq. (7) i.e., formulae for measurement updates.

$\hat{x}_k$ *:* Estimation of a posteriori state (state Correction) and
P_k: Estimation of a posterior error (*Error Covariance correction)* calculated by Eqs. (8) and (9), with measurement z_k.
The time and measurement equations are iteratively computed using earlier a posteriori, which estimates to forecast fresh apriori estimates. One of the Kalman filter's most notable features is its recursive nature in estimating states.

2.1.2. Kalman Filter for object detection and Multi-object Tracking to predict very nearer objects

The Kalman filter is employed in Multi-Object Tracking (MOT), which encompasses the two distinct phases. The first stage is Object Detection, which uses a Bounding Box and its centroid position to identify the location and categories of the objects and assign them a unique instance ID. To detect and extract moving objects, used a background subtraction method (Gaussian model). And, also must assess if many objects are obstructing the view. The second stage is the tracking approach which utilizes feature information of bounding boxes and their centroid position across frames in the video. This information will be used to track the next state (or location) of the objects from frame to frame in the video. While tracking, find the distance between a centroid of the bounding box to another centroid of the bounding box in the video frames. The geometric features of an object can be described in terms of position, shape, center of mass (centroid), etc.
The suggested methodological implementation utilizes the UCF-101 public dataset, which comprises 90-second videos sampled at 24 frames per second. For the experiment, Figure 3 depicted videos 1 to 6 has been utilized to predict the frame in which two humans are very near to each other. Kalman Filter uses the object's position from the first frame, depicted in Figure 3, as input to subsequently track the object in the following frames. After the object is detected, precise localization makes it possible to refine the regions and encapsulate the object using bounding boxes **(**tracking window) and the centroid position of the object.

Figure 3. Videos 1 to 6 of the UCF-101 dataset are considered for implementation

A. Object Detection by using a Bounding Box and its centroid position

A Bounding Box (BB) is a rectangle that surrounds an object and determines its position. BBs are utilized in the task of object detection, which aims to determine the position and kind of several objects in a picture. The main parameters that define a bounding box are: 1) (x_1, y_1): denotes the x and y coordinates of the rectangle's upper left corner 2) (x_2, y_2): denotes the x and y coordinates of the rectangle's bottom right corner 3) (x_c, y_c): denotes the x and y coordinates of the bounding box's center, as computed by Eqs. (10) and (11) 4) Width: represents the width of the bounding box calculated by Eq. (12) 5) Height: represents the height of the bounding box calculated by using Eq. (13) 6) Labels of the BBs (ID -Identifier). The representation forms of BB are as follows:

$$x_c = \frac{(x_1+x_2)}{2} \quad (10)$$

$$y_c = \frac{(y_1+y_2)}{2} \quad (11)$$

$$w = (x_2 - x_1) \quad (12)$$

$$h = (y_2 - y_1) \quad (13)$$

The Bounding Box's centroid has the x-coordinate x_c, and the y-coordinate y_c, w is the width of the BB, and h is the height of the BB. The centroid position of the object (or BB) is represented by (x_c, y_c), as illustrated in Figure 4 and also depicted in Figure 5.

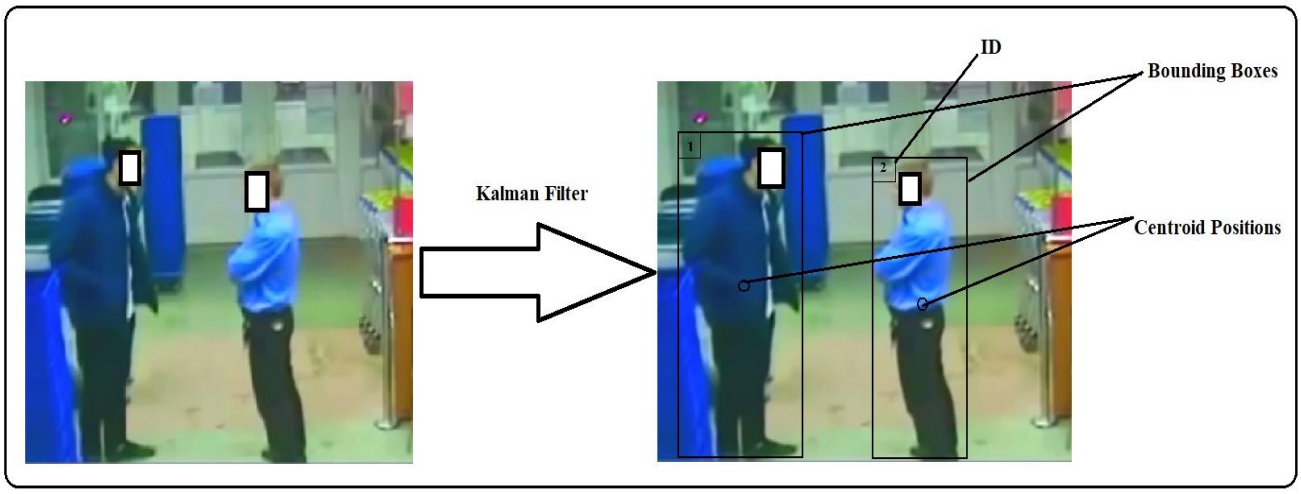

Figure 4. Kalman filter-based Human Tracking using Bounding Box and centroid position

B. Multi-object Tracking to predict very nearer objects

Object tracking is concerned with image and video sequences. It is an extension of object detection in which one or more objects are identified in a series of images. In Multiple Object Tracking (MOT), instance IDs are assigned to multiple objects, as demonstrated in Figures 4 and 5. Consequently, there is minimal variation in the position of moving objects between two consecutive frames. As a result, the size of the tracking window (bounding box) and the centroid of the moving target exhibit slight changes between the frames. Figure 5 shows that bounding boxes (segments) for numerous objects must be input at the initial frame. Therefore, to describe a moving object, choose the centroid and tracking window size as feature values.

To monitor the segmented moving objects and minimize noise, initially assign a tracking window to each object in the scene. The tracking window size ought to be marginally larger than the size of the object image. It can thereby increase operational speed while decreasing noise interference and picture processing time. The three submodules of the Kalman filter tracking model are the i) motion model ii) feature matching, and iii) model update.

i) Motion Estimation Model

The motion model predicts the object's future position based on the prior position data collection. The Kalman filter, used for tracking, is described by its states, motion model, and measurement equation matrix $[X_k]$, which are defined in Eq. (14). The system state vector has eight dimensions and can be expressed as follows:

$$X_k = [x_{0,k}, y_{0,k}, l_k, h_k, v_{x,k}, v_{y,k}, v_{l,k}, v_{h,k}]^T \quad (14)$$

Where, $x_{0,k}$, $y_{0,k}$ represent horizontal and vertical centroid coordinates, l_k, h_k indicate tracking window's half-width and half-height, and $v_{x,k}$, $v_{y,k}$, $v_{l,k}$, $v_{h,k}$ signifies their respective speeds. The system measurement vector z_k is defined in Eq. (15) and adopts the following form:

$$Z_k = [x_{0,k}, y_{0,k}, l_k, h_k]^T \quad (15)$$

Figure 5. Bounding boxes and centroid position of the humans of initial frames in videos 1 to 6.

The following includes the tracking system's transition matrix (A) and measurement matrix (H), as well as the Gaussian process w_k and measurement v_k. The noise settings depend entirely on the system being tracked and modified empirically.

$$\begin{bmatrix} 1 & 0 & 0 & 0 & \Delta t & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & \Delta t & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & \Delta t & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & \Delta t \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

The observation matrix H is defined as follows:

$$\text{H}= \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix}$$

The video has track-to-track (Δt) intervals. After the motion model's state equation and measurement equation have been defined, the Kalman filter can be used in the following frame to ascertain the object's position and size within a limited range, as well as to acquire trajectories of moving objects.

ii) Feature matching

Every moving object is characterized by its centroid and tracking window. The coordinates of the horizontal and vertical centroids, along with the area of the i^{th} object in the k^{th} frame, are specified as x_k^i,y_k^i, and S_k^i.

Eq. (16) is used to determine the centroid distance function ($D(i,j)$) between the i^{th} object in the k^{th} frame and the j^{th} object in the $k+1^{th}$ frame.

$$D(i,j) = \frac{|\sqrt{\left(x_k^i - x_{k+1}^j\right)^2 + \left(y_k^i - y_{k+1}^j\right)^2}|}{Max_n\,|\sqrt{\left(x_k^i - x_{k+1}^n\right)^2 + \left(y_k^i - y_{k+1}^n\right)^2}|} \qquad (16)$$

Eq. (17) calculates the area ($A(i,j)$) difference between the i^{th} item in the k^{th} frame and the j^{th} object in the $k+1^{th}$ frame, which is defined as:

$$A(i,j) = \frac{|S_k^i - S_{k+1}^j|}{Max_n|S_k^n - S_{k+1}^n|} \qquad (17)$$

$S_k^i = 4\, l_k^i h_k^i$ indicates the tracking window's area. It shows how much the window has deformed; the lower the number, the more accurately the shapes of the two objects are described. Using these definitions, a cost function is defined as shown in Eq. (18).

$$V(i,j) = \alpha D(i,j) + \beta A(i,j) \qquad (18)$$

α, β are track filtering or update equations parameters. Here, $\alpha = 0.8$, $\beta = 0.2$ where $\alpha + \beta = 1$. A lower value of the cost function indicates a higher likelihood of correspondence between the two objects. Upon identification, we identified that many object sections are interconnected, as illustrated in Figure 6, video-5. The areas of the objects have fused, allowing to track many objects as a single object as well as creating a new object-matching eigenvalue. If an object has several moving objects, first split it into a large number of distinct moving objects to see if the merging has already taken place. If it has, then match the split objects 'with the objects' features before dividing. If not, we split the objects, assign new tracking windows to track moving objects, and create new eigenvalues, treating them as new.

iii) Model Update

After determining the cost function's minimum value, update the Kalman filter motion model's parameters using the $k + 1^{th}$ frame characteristics, which are then used as input in the following frame. Continuously doing this process in the number of iterations to complete the model update till the moving objects vanished.

So finally, Table 1 reported the bounding box IDs, centroid, and area distance values i.e., D (i, j), A (i, j), and the cost function (V) values of the initial, and predicted frames of the videos 1-6 of Figure 6. Figure 6 presents predicted frames of videos 1 to 6 which have KF smaller cost function (V) value, and also humans are very near to each other. Video-1: Frame No-1098, Video-2: Frame No-992, Video-3: Frame No-600, Video-4: Frame No-2025, Video-5: Frame No-1566, and Video-6: Frame No-1231 are predicted frames that have smaller cost function value. The computational complexity of a Kalman filter algorithm for tracking objects in a video, utilizing a cost function, regarded as $O(N^2)$, where 'n' denotes the dimension of the estimated state vector, due to the primary operations involving matrix multiplications and inversions that scale quadratically with the state size.

Table 1. Cost function values of initial and predicted frames of the videos 1 to 6

Video No	Type of the Frame	Bounding Boxes IDs	Centroid Distance D (i, j) In meters	Area A (i, j) In meters	Cost Function Value V (i, j)
1	Initial Frame	1, 2	0.6	0.9	0.66
	Predicted Frame	1, 2	0.2	0.1	0.18
2	Initial Frame	1, 2,3, 4	0.9	0.9	0.9
	Predicted Frame	1, 2	0.3	0.1	0.26
3	Initial Frame	1, 2	0.5	0.4	0.48
	Predicted Frame	1, 2	0.1	0.1	0.1
4	Initial Frame	1, 2	0.9	0.9	0.9
	Predicted Frame	1, 2	0.3	0.2	0.28
5	Initial Frame	1, 2, 3, 4, 5, 6	0.7	0.8	0.72
	Predicted Frame	1, 2	0.4	0.3	0.38
6	Initial Frame	1, 2	0.9	0.9	0.9
	Predicted Frame	1, 2	0.2	0.3	0.22

Figure 6. Predicted frames of Video 1 to 6, Video-1: Frame No-1098, Video-2: Frame No-992, Video-3: Frame No-600, Video-4: Frame No-2025, Video-5: Frame No-1566, and Video-6: Frame No-1231 have smaller cost function values.

2.2. Step-2: MediaPipe Pose (MPP) Landmark Detector Feature Extraction Technique

MPP landmark is used to identify significant human body parts in a picture or video. It accepts an image or frame as input and generates the location of each instance's posture landmark. It also calculates the visibility (v) value and the placements of (x- image width), (y- image height), and (z- landmarks depth) (3D) lie between 0.0 and 1.0 for one and all-important spots [24].

Figure 6 presented KFA predicted video frame numbers- Video-1: Frame No-1098, Video-2: Frame No-992, Video-3: Frame No-600, Video-4: Frame No-2025, Video-5: Frame No-1566, and Video-6: Frame No-1231 has been considered as input to MediaPipe Framework to identify landmarks and postures. Figure 7 displays predicted frame numbers: 1098 designated as P1, 992 as P2, 600 as P3, 2025 as P4, 1566 as P5, and 1231 as P6 respectively. Figure 8 also presents templates for shooting, fighting, and kicking activities.

To construct landmarks of Figures 7 and 8 of humans, the MediaPipe framework [15] is used. MediaPipe Pose [9] Framework is used to find the landmarks (key body locations), and to analyze the posture of the human body [10], [21], [22]. Human landmarks [21] (0 to 32) are depicted in Figure 9. Landmarks and postures are extracted by using a MediaPipe Pose (MPP) landmark detector.

Figure 7 presents predicted frames (P1 to P6) and Figure 8 presents templates (shooting, fighting, and kicking) that are passed to the MPP detection ML pipeline, which has 2-step detectors i) Person Detector and ii) Landmarks Detector. MediaPipe features an integrated automatic pre-processing mechanism for video data that decreases computational complexity relative to manual implementation. The MP library employs an integrated automatic selfie segmentation pre-processing model [25]. It uses a process function to remove the background of the video frame. Human landmarks and posture in the video frame are estimated. MP estimates the human objects' landmarks (0 to 32), in the pre-processed video frames as shown in Figure 9.

Figure 7. Predicted Video frames (P1 to P6) of different human activities from left to right (train data)

Figure 8. Shooting, Fighting, and Kicking activities to test (templates) from left to right.

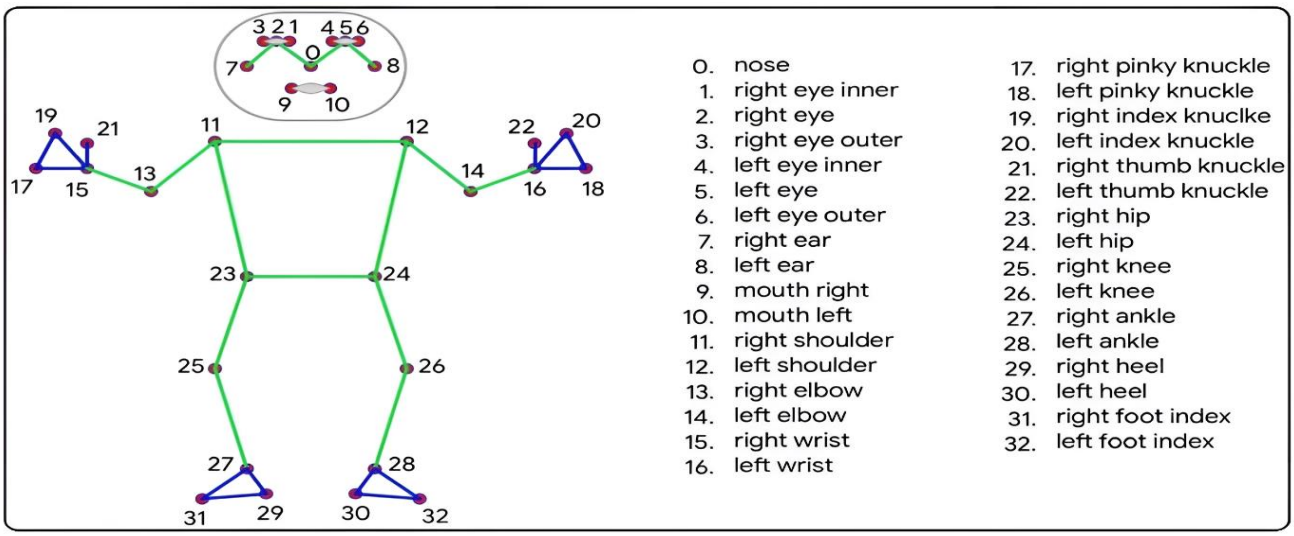

Figure 9. Pose reference points from Mediapipe BlazePose [21]

The pose detection model [26] is used to detect the presence of bodies with key pose landmarks and the postures of the training data (P1 to P6) and test data (templates), as shown in Figures 10 and, 11 respectively. The MediaPipe Pose (MPP) detection model effectively predicts landmarks in diverse video conditions, including low quality, occlusion, overlapped people, small objects, night videos, moving objects, etc. MPP default complexity is 1.

Figure 10. Postures (P1 to P6) of different human activities from left to right (train data)

Figure 11. Postures of Templates Shooting, Fighting, and Kicking activities to test (templates) from left to right

As shown in Figure 9, 0 to 32 expected landmarks for the human stance, six important key (or feature vectors) related to the human posture are used to recognize human activity. Those are 1) 16. Left_Wrist 2) 15. Right_Wrist 3) 24. Left_Hip 4) 23. Right_Hip 5) 28. Left_Ankle 6) 29. Right_Ankle. The movements of a person's hands and legs are used to identify human activity. Figure 12 illustrates how we can calculate the pairwise distance or angle between two landmarks. To quantify hand and leg movements, pairwise distance values are employed.

Table 2 presents 10 different combinations of pairs of landmarks that are used to recognize human activity. Pairwise distances between landmarks are calculated using the Euclidean distance (d) metric.

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \qquad (19)$$

Euclidean distance between pairs of landmarks of training data and test data (templates) are calculated using Eq. (19). The calculated results are presented in Table 3. In addition, the Euclidean distance

between pairs of landmarks of the test data (templates) was calculated using Eq. (19). The calculated values are presented in Table 4.

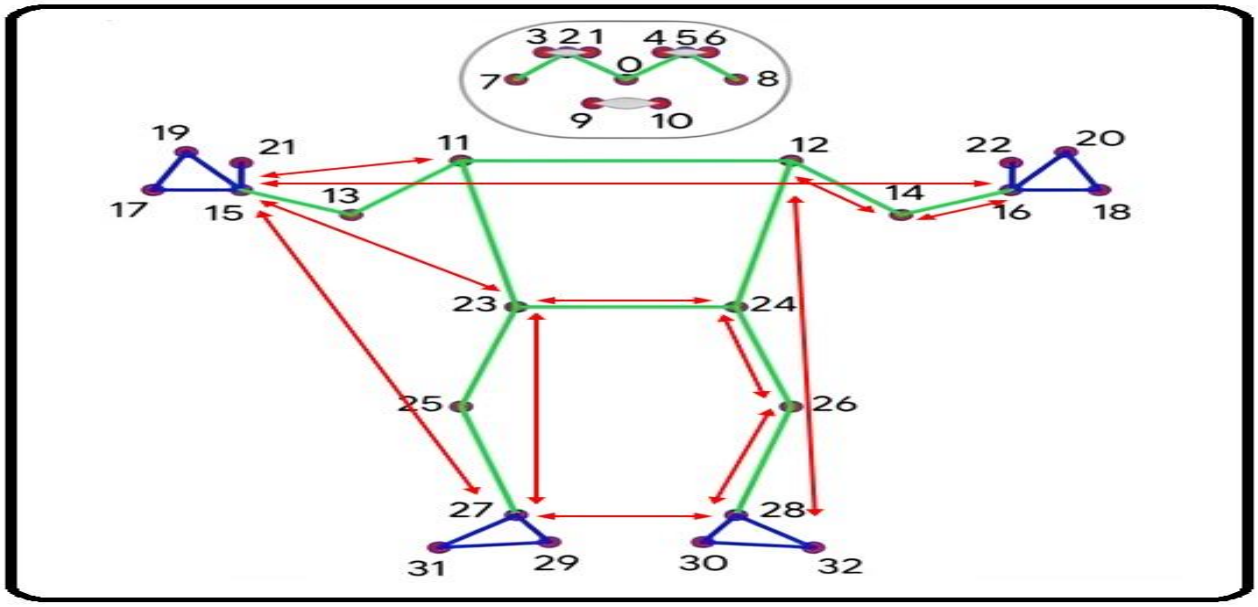

Figure 12. Principal pairwise distances are used in the feature vector representing the position

Table 2. 10 different combinations of pairs of landmarks (D_1 TO D_{10})

SNO	Pairs of Landmarks	Name of the Distances
1	(Left_Wrist, Right_Wrist)	D_1= EucDist (Left_Wrist, Right_Wrist)
2	(Left_Hip, Left_Wrist)	D_2= EucDist (Left_Hip, Left_Wrist)
3	(Right_Hip, Right_Wrist)	D_3= EucDist (Right_Hip, Right_Wrist)
4	(Left_Hip, Right_Wrist)	D_4= EucDist (Left_Hip, Right_Wrist)
5	(Right_Hip, Left_Wrist)	D_5= EucDist (Right_Hip, Left_Wrist)
6	(Left_Ankle, Right_Ankle)	D_6= EucDist (Left_Ankle, Right_Ankle)
7	(Left_Ankle, Left_Wrist)	D_7= EucDist (Left_Ankle, Left_Wrist)
8	(Right_Ankle, Right_Wrist)	D_8= EucDist (Right_Ankle, Right_Wrist)
9	(Left_Ankle, Right_Wrist)	D_9= EucDist (Left_Ankle, Right_Wrist)
10	(Right_Ankle, Left_Wrist)	D_{10}= EucDist (Right_Ankle, Left_Wrist)

Table 3. D_1 to D_{10} Euclidean Distances between pairs of Landmarks of Train data vectors (Video Frames)

SNO	Name of the Postures (Train Data)	D_1(m)	D_2(m)	D_3(m)	D_4(m)	D_5(m)	D_6(m)	D_7(m)	D_8(m)	D_9(m)	D_{10}(m)
1	Posture P1	0.0196	0.0744	0.1019	0.0643	0.1161	0.1740	0.3020	0.3906	0.3052	0.3906
2	Posture P2	0.1213	0.1262	0.1008	0.1329	0.1448	0.0736	0.3585	0.3706	0.3622	0.3706
3	Posture P3	0.0001	0.0427	0.0204	0.0427	0.0203	0.1162	0.1214	0.1288	0.1212	0.1288
4	Posture P4	0.0001	0.1996	0.1743	0.1995	0.1744	0.1610	0.4570	0.3999	0.4570	0.3999
5	Posture P5	0.1546	0.1170	0.1750	0.2298	0.1147	0.2296	0.4175	0.4379	0.4280	0.4379
6	Posture P6	0.1561	0.3553	0.4466	0.4789	0.3261	0.3461	0.6680	0.9288	0.7130	0.9288

Table 4. D_1 to D_{10} Euclidean Distances between pairs of Landmarks of Test data vectors (Templates)

SNO	Name of the Postures (Train Data)	D_1(m)	D_2(m)	D_3(m)	D_4(m)	D_5(m)	D_6(m)	D_7(m)	D_8(m)	D_9(m)	D_{10}(m)

1	Shooting Activity Template Posture	0.0146	0.3189	0.3284	0.3200	0.3292	0.1252	0.4611	0.5514	0.4631	0.5514
2	Fighting Activity Template Posture	0.0415	0.2550	0.2895	0.2701	0.2650	0.0933	0.5645	0.5436	0.5781	0.5436
3	Kicking Activity Template Posture	0.0768	0.1983	0.2906	0.2342	0.2370	0.2781	0.3687	0.5877	0.3656	0.5877

2.3. Step-3: Templates Matching Statistical Approaches (TMSA) - based Classification

Template matching is an advanced machine vision technique utilized for the identification of segments within an image that exhibit similarity to a reference template (patch). The major component is the Source image (I), which is the image that is anticipated to match with the template image. The template picture (T) is the patch image designated for comparison with the source image. This paper uses three types of Templates Matching Statistical Approaches (TMSA) to find the similarity between train data vectors (P_1 to P_6 postures) and test data (template) vectors of shooting, fighting, and kicking activities. To predict human crime activity, Cross Correlation (CC), Normalized Cross Correlation (NCC), and Sum of Absolute Difference (SAD) are used.

2.3.1. Cross Correlation (CC)

Cross-correlation evaluates the similarity between two sequences based on their relative displacement using Eq. (20).

$$R_{xy} = \frac{\sum_{i=1}^{n}(x_i-\bar{x})\,(y_i-\bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i-\bar{x})\,(y_i-\overline{\bar{y}})}} \qquad (20)$$

Consider the two series x(i) and y(i), with i=1, 2, ...N. The cross-correlation function R. $\bar{x}$ and $\bar{y}$ are the means of the corresponding series. The value ranges from -1 to 1: a value of 1 signifies that the data sets completely coincide (similar to perfectly aligned combs), 0 indicates that there is no correlation, and -1 indicates that they are opposed. The computational complexity of calculating a cross-correlation function based on relative displacement between two signals is typically O(N), where N is the number of data points.

2.3.2. Normalized Cross Correlation (NCC)

A mathematical method for calculating the linear relationship between two variables is the normalized cross-correlation (NCC). This is a specialized form of cross-correlation that is normalized to achieve scale invariance, meaning it remains unaffected by variations in the overall intensity of the signals. Consequently, it serves as a more reliable metric of similarity; specifically, it is the cross-correlation value adjusted to a range of -1 to 1, where 1 denotes perfect correlation and -1 signifies perfect anti-correlation. The computation is exceptionally rapid and efficient. The most basic representation of normalized cross-correlation (NCC) is the cosine of the angle θ between two vectors x and y. Consider two series x(i) and y(i) where i=1,2...N. The distance function $NCCR_{xy}$ determines how near two coordinates are using Eq. (21).

$$NCCR_{xy} = \cos\theta = \frac{\sum_{i=1}^{n}(x_i) * (y_i)}{\sqrt{\sum_{i=1}^{n}(x_i) * (y_i)}}$$

$$-1 \leq NCC \leq 1 \qquad (21)$$

The computational complexity of the Normalized Cross Correlation (NCC) function is typically considered to be O(MN), where M and N represent the dimensions or vectors of the two images being compared, i.e., computation time scales linearly with the number of pixels in the images when calculated directly. The training vectors size M×N and a template vector size m×n, then the computational complexity of the NCC involves 3·m·n·M·N additions/subtractions and 2·m·n·M·N multiplications.

2.3.3. Sum of Absolute Difference (SAD)

The similarity between image blocks is measured by the SAD. Eq. (22) is used to compute the absolute difference between each pixel in the original block (S_1-Subset1) and the corresponding pixel in the block (S_2- Subset 2) being used for comparison. A measure of block similarity is produced by adding the differences. Consider two series x(i) and y(i) where i=0,1,2...N-1.

$$SAD = \sum_{x=0}^{x-1} \sum_{y=0}^{y-1} |S_1[x,y] - S_2[x,y]|. \qquad (22)$$

The total of the absolute differences between an element and every other element is determined by the array's location and is obtained from the sorted data inputs. This permits an O(n) linear time complexity.

Eqs. 20, 21, and 22 are used to calculate the similarity between Table 3 and 4 mentioned train data vectors (Posture P1 to P6) and test data vectors (shooting, fighting, and kicking activities), to predict human activity. The results are depicted in Table 5. From three of the TMSA similarity measure values, the NCC has the highest similarity measure value to predict shooting, fighting, and kicking crime activities in humans.

Therefore, the use of NCC calculation is efficient and highly recommended to statistically evaluate the similarity between datasets. So, the computation complexity for human crime activity prediction (HAP) using the NovelHAP algorithm with KFA, MPP, and TMSA is quadratic., i.e., $O(N^2)$.

Table 5. Human Activity Prediction Results

Similarity between Train data & Test data	Cross-Correlation (CC)	Normalized Cross-Correlation (NCC)	Sum of Absolute Difference (SAD)	Result of Human Activity Prediction
Similarity between (Posture P1 to P6) & (Shooting Activity Posture Template)				
*Posture P1 & Shooting Activity Posture Template	**0.9258**	****0.9570**	**0.9136**	**Shooting Activity Predicted at Posture P1 with highest NCC similarity is 0.9570**
Posture P2 & Shooting Activity Posture Template	0.8839	0.9393	0.8993	---
Posture P3 & Shooting Activity Posture Template	0.6151	0.8948	0.6457	---
Posture P4 & Shooting Activity Posture Template	0.7132	0.8652	0.6953	---
Posture P5 & Shooting Activity Posture Template	0.7252	0.8228	0.7324	---
Posture P6 & Shooting Activity Posture Template	0.8345	0.853	0.8034	---

Similarity between (Posture P1 to P6) & (Fighting Activity Posture Template)				
Posture P1 & Fighting Activity Posture Template	0.7369	0.8698	0.8591	---
Posture P2 & Fighting Activity Posture Template	0.7925	0.8253	0.8863	---
Posture P3 & Fighting Activity Posture Template	0.8953	0.9146	0.9283	---
*Posture P4 & Fighting Activity Posture Template	**0.9682**	****0.9924**	**0.9596**	**Fighting Activity Predicted at Posture P4 with highest NCC similarity is 0.9924**
Posture P5 & Fighting Activity Posture Template	0.8564	0.8965	0.8564	---
Posture P6 & Fighting Activity Posture Template	0.8938	0.8325	0.8821	---
Similarity between (Posture P1 to P6) & (Kicking Activity Posture Template)				
Posture P1 & Kicking Activity Posture Template	0.7653	0.9035	0.8145	---
Posture P2 & Kicking Activity Posture Template	0.7012	0.8969	0.8007	---
Posture P3 & Kicking Activity Posture Template	0.8069	0.9136	0.8267	---
Posture P4 & Kicking Activity Posture Template	0.8048	0.8835	0.8159	---
Posture P5 & Kicking Activity Posture Template	0.9253	0.9692	0.8973	---
*Posture P6 & Kicking Activity Posture Template	**0.9630**	****0.9931**	**0.9630**	**Kicking Activity Predicted at Posture P6 with the highest NCC similarity is 0.9931**

3. Experimental Results and Discussion

This section delineates the dataset specifications, experimental outcomes on the dataset, and empirical evaluation. The proposed methodology was executed and evaluated in different devices, and platforms (Windows and Linux) using the UCF-101 real-world surveillance dataset. The results obtained are examined in detail. Finally, the proposed method is compared against the existing methods.

3.1. Results

The proposed methodology is executed and evaluated on the UCF-101 public dataset for real-world surveillance crime. The dataset comprises 950 videos in total. The experiment employed a video duration of 90 seconds and a frame rate of 24 frames per second. The 13 real-world anomalies captured by the camera included abuse, arrests, arson, assault, car crashes, burglaries, explosions, fights, robberies, shootings, thieving, shoplifting, and vandalism. The dataset was utilized to examine human crime activity prediction (HAP) in extensive unedited surveillance video datasets (indoor and outdoor), addressing significant video complexity challenges. Figure 13 illustrates several video obstacles arranged from left to right 1) Occlusion, cluttered background, low-resolution, illumination, all persons are not exactly facing the camera 2) Low video and image quality, overlapped humans, moving humans, small objects (humans), different poses 3) People wearing different clothes, people laid down on the floor, night videos, different weather conditions.

Figure 13. For the experiment UCF-101 CCTV footage surveillance video frames obstacles from left to right

The proposed methodology was evaluated using A) a limited dataset of 200 videos and B) an extensive dataset of 950 videos for human activity prediction.

A) the small number of data samples: 200 videos have been utilized for Human Crime Activity prediction (HAP) encompassing shooting, fighting, and kicking. The novel HAP algorithm with the Kalman Filter (KF) algorithm, MPP framework, and Template Matching Statistical Approaches (TMSA) were used to analyze a dataset that included 100 videos of criminal conduct and 100 videos of non-criminal activity. The obtained True Positive (TP) is 96, True Negative (TN) is 94, False Positive (FP) is 6, and False Negative (FN) is 4. The algorithm's performance is evaluated by the TPR (True Positive Rate), TNR (True Negative Rate), FPR (False Positive Rate), FNR (False Negative Rate), Accuracy, Precision, Recall, and F_1-Score. The obtained TPR is 96%, the FPR is 6%, the FNR is 4%, the TNR is 94%, the accuracy is 95%, the precision is 94.11%, the recall is 96%, and the F1-Score is 95%. The findings of the confusion metrics and performance measurements are presented in Table 6.

B) a large number of data samples: 950 videos have been considered for Human Crime Activity prediction (HAP) encompassing shooting, fighting, and kicking. A dataset with 475 crime activities and 475 no-crime activities videos was considered and applied the NovelHAP algorithm with KFF, MPP, and TMSA. The obtained TP is 460, TN is 462, FP is 13, and FN is 15. The obtained TPR is 96%, the FPR is 3%, the FNR is 4%, the TNR is 96.53%, the accuracy is 96.01%, the precision is 96.51%, the recall is 96%, and the F1-Score is 96.25%. Table 6 reports the outcomes of performance measures and confusion metrics.

Table 6. Results of the NovelHAP on the UCF-101 dataset using KF, MPP, and TMSA

Dataset	No. of Video	Samples in Each Class	TP	FP	TN	FN	TPR %	TNR %	FPR %	FNR %	Accuracy %	Precision %	Recall %	F1-Score %
Tested on: Small number of data samples			Human Crime Activity Prediction (HAP)											
NovelHAP Algorithm: KF, MPP, and TMSA														
		Crime Activities	No Crime Activities											

UCF-101	**200 Videos**	**100**	**100**	**96**	**6**	**94**	**4**	**96**	**94**	**6**	**4**	**95**	**94.11**	**96**	**95**
Tested on: Large number of data samples								**Human Crime Activity Prediction (HAP)**							
NovelHAP Algorithms: KF, MPP, and TMSA															
		Crime Activities	**No Crime Activities**												
UCF-101	**950 Videos**	**475**	**475**	**460**	**13**	**462**	**15**	**96**	**96.53**	**3**	**4**	**96.01**	**96.51**	**96**	**96.25**

The experimental findings indicated that the proposed NovelHAP algorithm with KFA, MPP, and TMSA for Human Crime Activity Prediction (HAP), was executed and evaluated on the UCF-101 real-world surveillance benchmark dataset, which includes both small and big data samples, yielded optimal performance in a single implementation. Compared to other filtering techniques, such as particle filters, a Kalman filter has a less computational cost and is suited for real-time applications because of its recursive nature of calculations based on the current state and measurement. KFA has a quadratic computational complexity, or $O(n^2)$.

The MediaPipe Pose (MPP) framework reduces computing costs and does away with the need for manual implementation by integrating an automatic pre-processing technique for video data. Additionally, in a variety of intricate scenarios, MPP can predict human landmarks when they overlap with other things, etc., with constant time. In the NovelHAP algorithm's process, KFA initially identifies the closest humans in the video frames to one another. The experiment uses 950 videos, each having a duration of 90 seconds, the calculation for a single video yields 90 seconds multiplied by 24 frames per second, resulting in 2,160 frames. For 950 videos, the total is 2,160 frames multiplied by 950, equating to 20,52,000 video frames. KFA extracted a single frame from a single video, depicting persons near one another. Ultimately, out of 20,52,000 video frames, only 950 frames were transferred to MPP (phase-2). The MPP predicts one human posture and the landmarks (0–32) of each frame. The Euclidean distance from D_1 to D_{10} for training and test vectors is determined using six key landmarks. The dimensionality reduction has been done. TMSA calculated the similarity between sparse data. Consequently, the computational complexity is quadratic, or $O(N^2)$.

Additionally, the suggested methodology was successfully deployed across several platforms and devices, with repairs and updates conducted as issues emerged. Interoperability, compatibility, reliability, and maintainability have been accomplished using standard methodologies KFA, and TMSA in the NovelHAP algorithm. The efficacy of Deep Learning (DL) algorithms diminishes when executed with a small number of data samples. The optimal result obtained is 95% utilizing the UCF-101 dataset, as this study shows, even with small data sets. The optimal result achieved with the diminished computational expense for Human Activity prediction (HAP) is 96.01% utilizing the UCF-101 dataset, as demonstrated in this work with extensive datasets. There have been few false positives and false negatives in this study's analysis of videos on a variety of complex scenarios where machine learning (ML) techniques are less effective. Therefore, the computational complexity of the NovelHAP method for the prediction of human crime activity is quadratic, i.e., $O(N^2)$.

3.2. Comparison with Existing Methods

A literature survey was conducted to predict human crime activities such as shooting, fighting, and kicking in videos from various datasets using various algorithms and methodologies. [16] proposed an architecture for video activity prediction using Saliency-aware Motion Enhancement (SME) with a

Weighted Long Short-Term Memory Network (WLSTM). The framework achieved the highest prediction results of 98.3% on the UT-Interaction dataset set 1. [17] used a Bi-LSTM classifier with two tri-axial IMU sensors and predicted future activities. The proposed HAP system predicts future actions with an average accuracy of 97.96%. [8] proposed multi-network fusion (ConvNet and LiteFlowNet) technique for detecting violence or the absence of violence in real-world surveillance. The Hockey Fight dataset achieves the highest level of accuracy at 98.62%.
[14] proposed CNN with an Improved Tree Seed algorithm for human activity recognition. The UT-Interaction dataset produces the highest accuracy of 100%. [15] offer the ConvLSTM method, LRCN algorithm, and CNN model to recognize human activities. The CNN model used the UCF50 dataset and achieved the highest accuracy of 99.58%. [10] used the classifiers Gradient Boosting (GB), Random Forest (RF), Ridge Regression (RR), and Logistic Regression (LR) to recognize the namaste, standing, bending down, and raising both hands, types of activities with 98% accuracy using the customized dataset. [20] used a novel AM-DLFC model for HAR, and utilizes CNN to recognize run, pick, and sit-up activities with 97% accuracy using the KU-HAR dataset. [23] presented a unique fusion approach 2s-AGCN. The public dataset FMS was used to recognize human posture with a higher accuracy of 98.2%.

Table 7. Comparison of proposed methodology with existing methodologies

Ref.	Year	Algorithms	Dataset/ Sensor Units	Outcome	Accuracy/Average Error Rate
Proposed Methodology	-	NovelHAP: KFA, TMSA	UCF-101 Small data samples	HAP: Shooting, Fighting, Kicking	95%
			UCF-101 Large data samples		96.01%
[8]	2024	ConvNet + LiteFlowNet)	RWF-2000	Human Violence and Non-Violence Detection	94.50%
			Surveillance Fight		96.69%
			Hockey Fight		98.62%
[15]	2024	CNN	UCF50 Dataset	Human Activity Recognition (HAR)	99.58%
[23]	2024	2s-AGNN	FMS Dataset	HAR	98.2%
[17]	2023	Bi-LSTM	2 tri-axial IMU sensors	Predict future actions	97.96%
[10]	2023	GB, RF, RR, LR	Customized Dataset	HAR	98%
[14]	2023	CNN + Improved Seed	UT-Interaction Dataset	HAR	100%
[20]	2023	AM-DLFC	KU-HAR Dataset	HAR: run, pick, sit up	97%
[16]	2021	SME, WLSTM	UT-Interaction Dataset	HAP	98.3%

Table 7 compares the proposed approaches with existing methodologies for the prediction of human criminal actions when humans are very near to each other in the video frames. The comparison results demonstrated that the NovelHAP proposed method yields optimizable outcomes for shooting, fighting, and kicking utilizing the methods Kalman Filter Algorithm (KFA), and Template Matching Statistical Approaches (TMSA). Furthermore, it has been demonstrated that the suggested methods are in optimizable computational costs in a variety of complex video scenarios and on both small and big data samples taken from the dataset.

4. Conclusion and Future Scope

In many complex contexts, it is difficult to predict human criminal action in surveillance video frames. For human crime activity prediction (HAP), the suggested NovelHAP algorithm used a predicted correction filter algorithm, skeleton-based method, and pattern-matching techniques such as the Kalman Filter Algorithm (KFA), MediaPipe Pose (MPP) framework, and Template Matching Statistical Approaches (TMSA). KFA predicted a video frame with a near human Bounding Box (BB)

and its centroid position compared to another. Human action can be expected when two persons are close. The anticipated frames were taken into consideration and identified the human's posture and landmarks using the MediaPipe Pose (MPP) Framework. The Euclidean distances are calculated from D_1 to D_{10} among the six significant landmarks: Left_Wrist, Right_Wrist, Left_Hip, Right_Hip, Left_Ankle, and Right_Ankle, of video frame postures (training vectors) and template vectors (test vectors). Cross-correlation (CC), Normalized Cross Correlation (NCC), and Sum of Absolute Difference (SAD) template matching statistical methods (TMSA) were utilized to assess the similarity between trained and test vectors. Among the three measured values of TMSA similarity, the highest value is considered and predicted shooting, fighting, and kicking criminal activities in humans.

The empirical analysis demonstrated that the accuracies for HAP on the UCF-101 are 95% for short data samples and 96.01% for large data samples, respectively. The prediction of human criminal action in video frames has a quadratic computational complexity of $O(N^2)$. Nevertheless, this research may eventually broaden to encompass the following: i) the prediction of human crime actions such as beating, dragging, and so on in video frames, and ii) the recognition and prediction of human activity in multiple persons in a single scenario.

References

[1] Neil Shah., Nandish Bhagat., & Manan Shah. (2021). Crime forecasting: a machine learning and computer vision approach to crime prediction and prevention. Visual Computing for Industry. Biomedicine, and Art. 4(9):1-14. https://doi.org/10.1186/s42492-021-00075-z.

[2] Neziha Jaouedi., Francisco J. Perales., José Maria Buades., Noureddine Boujnah., & Med Salim Bouhlel. (2020). Prediction of Human Activities Based on a New Structure of Skeleton Features and Deep Learning Model. *Electronics*. 20(17):1-15. https://doi.org/10.3390/s20174944**.**

[3] Afef Salhi., Fahmi Ghozzi., & Ahmed Fakhfakh. (2020). Estimation for Motion in Tracking and Detection Objects with Kalman Filter. *Dynamic Data Assimilation – Beating the Uncertainties.* 1-16. https://doi.org/10.5772/intechopen.92863.

[4] Dah-Jing Jwo., & Amita Biswal. (2023). Implementation and Performance Analysis of Kalman Filters with Consistency Validation. *Mathematics.*11(3):1-19. https://doi.org/10.3390/math11030521.

[5] Li Hongmei., Huang Lin., Zhang Ruiqiang., Lv Lei., Wang Diangang., & Li Jiazhou. (2020). Object Tracking in Video Sequence based on Kalman filter. *International Conference on Computer Engineering and Intelligent Control (ICCEIC).* 1-5. http://dx.doi.org/10.1109/ICCEIC51584.2020.00029.

[6] Arija Birze., Kaitlyn Regehr., & Cheryl Regehr. (2022). Workplace Trauma in a Digital Age: The Impact of Video Evidence of Violent Crime on Criminal Justice Professionals. *Journal of Interpersonal Violence*. 38(1):1654-1689. https://doi.org/10.1177/08862605221090571.

[7] Diana Nagpal., & Shikha Gupta. (2023). Human Activity Recognition and Prediction: Overview and Research Gaps. *IEEE 8th International Conference for Convergence in Technology (I2CT).* 1-6. http://dx.doi.org/10.1109/I2CT57861.2023.10126458.

[8] Fath, U Min Ullah., Zulfiqar Ahmad Khan., Sung Wook Baik., Estefania Talavera., Saeed Anwar., & Khan Muhammad. (2024). Dual deep learning network for abnormal action detection. *IEEE.* 1-8. https://doi.org/10.1109/AVSS61716.2024.10672568.

[9] Jong-Wook Kim., Jin-Young Choi., Eun-Ju Ha., & Jae-Ho Choi. (2023). Human Pose Estimation Using MediaPipe Pose and Optimization Method Based on a Humanoid Model. *Applied Sciences.*13(14):1-21. https://doi.org/10.3390/app13042700.

[10] Shibhangi Kedar., Shivkanya Doiphode., Ankita Gaikwad., Divya Kurumkar., & Swati Shekapure. (2023). Human activity identification using pose estimation. *Int. J. Res. Pub. Rev.* 4(3):1463-1469.
[11] Amaury Auguste., Wissam Kaddah., Marwa Elbouz., Ghislain Oudinet., & Ayman Alfalou. (2021). Behavioral Analysis and Individual Tracking Based on Kalman Filter: Application in an Urban Environment. *Sensors.* 21(21): 1-19. https://doi.org/10.3390/s21217234.
[12] Yuting Bai., Bin Yan., Chenguang Zhou., Tingli Su., & Xuebo Jin. (2023). State of art on state estimation: Kalman filter driven by machine learning. *Annual Reviews in Control.* 56; 1-12. https://doi.org/10.1016/j.arcontrol.2023.100909.
[13] Azadeh Emami., Majid Sarvi., & Saeed Asadi Bagloee. (2019). Using Kalman filter algorithm for short-term traffic flow prediction in a connected vehicle environment. *Journal of Modern Transportation*. 27; 222-232. https://doi.org/10.1007/s40534-019-0193-2.
[14] Tehseen, A., Sohail, K., Shaheryar, N., Muhammad, A. K., Ye, J. K., & Byoungchol, C. (2023). HRNetO: Human action Recognition Using Unified Deep Features Optimization Framework, Computers. *Materials & Continua.* 75(1):1089-1105. https://doi.org/10.32604/cmc.2023.034563.
[15] Md, A. U., Md, A. T., Muhammad, S. U., Chandan, D., Moumita, C., Souvik, P., Md,
M. I., Ansam, K., Ammar, A., & Sunil, A. (2024). Deep learning-based human activity recognition using CNN, ConvLSTM, and LRCN. *International Journal of Cognitive Computing in Engineering.* 5; 259-268. https://doi.org/10.1016/j.ijcce.2024.06.004.
[16] Zhengkui Weng., Wuzhao Li., & Zhipeng Jin. (2021). Human activity prediction using saliency-aware motion enhancement and weighted LSTM network. *EURASIP Journal on Image and Video Processing.* 3; 1-23. https://doi.org/10.1186/s13640-020-00544-0.
[17] Ismael Espinoza Jaramillo., Channabasava Chola., Jin-Gyun Jeong., Ji-Heon Oh, Hwanseok Jung., Jin-Hyuk Lee., Won Hee Lee., & Tae-Seong Kim. (2023). Human Activity Prediction Based on Forecasted IMU Activity Signals by Sequence-to-Sequence Deep Neural Networks. *Sensors.* 23(14):1-16. https://doi.org/10.3390/s23146491.
[18] Niek Tax. (2018). Human Activity Prediction in Smart Home Environments with LSTM Neural Networks, *International Conference on Intelligent Environments*. 40-47. https://doi.org/10.1109/IE.2018.00014.
[19] Yegang Du., Yuto Lim., &Yasuo Tan. (2019). A Novel Human Activity Recognition and Prediction in Smart Home Based on Interaction. *Sensors.* 19(20):1-16. https://doi.org/10.3390/s19204474.
[20] Morsheda Akter., Shafew Ansary., Md. Al-Masrur Khan., & Dongwan Kim. (2023). Human Activity Recognition Using Attention-Mechanism-Based Deep Learning Feature Combination. *Sensors.* 23(12):1-15. https://doi.org/10.3390/s23125715.
[21] Jiaji Liu., Xiaofang Mu., Zhenyu Liu., & Hao Li. (2023). Human skeleton behavior recognition model based on multi-object pose estimation with spatiotemporal semantics. *Machine Vision and Applications.* 34(44):1-13. https://doi.org/10.1007/s00138-023-01396-0.
[22] Jong-Wook Kim., Jin-Young Choi., Eun-Ju Ha., & Jae-Ho Choi. (2023). Human Pose Estimation Using Mediapipe Pose and Optimization Method Based on a Humanoid Model. *Applied Sciences.*13(4):1-21. https://doi.org/10.3390/app13042700.
[23] Yahong Xu., Shoulin Wei., & Jibin Yin. (2024). Optimization of Human Posture Recognition based on Multi-view Skeleton Data Fusion, *Information Technology Control*. 53(2):542-553. https://doi.org/10.5755/j01.itc.53.2.36044
[24] Ming-Hwa Sheu., S. M. Salahuddin Morsalin., Chung-Chian Hsu., Shin-Chi Lai., Szu-Hong Wang., & Chuan-Yu Chang. (2023). Improvement of Human Pose Estimation and Processing with the Intensive Feature Consistency Network. *IEEE.*1-15. https://doi.org/10.1109/ACCESS.2023.3258417.

[25] Yong-Woon Kim., Yung-Cheol Byun, Addapalli V. N. Krishna., & Balachandran Krishnan. (2021). Selfie Segmentation in Video Using N-Frames Ensemble. *IEEE*. 9; 163348-163362. http://doi.org/10.1109/ACCESS.2021.3133276.

[26] Nan Ma., Zhixuan Wu., Yiu-ming Cheung., Yuchen Guo., Yue Gao., Jiahong Li., & Beijyan Jiang. (2022). A Survey of Human Action Recognition and Posture Prediction. *Tsinghua Science and Technology*. 27(6): 973–1001. https://doi.org/10.26599/TST.2021.9010068.