IOSR Journal of Computer Engineering (IOSR-JCE)
e-ISSN: 2278-0661, p-ISSN: 2278-8727, Volume 27, Issue 5, Ser. 2 (Sept. – Oct. 2025), PP 01-18
www.iosrjournals.org

Motion Detection Based On Hybrid Modified GMM & KNN Algorithm

Tarek Salah Elhabian
The Higher Institute Of Computer And Information Technology, El Shorouk Academy, Cairo, Egypt

Abstract:
Accurate detection of moving objects is a fundamental challenge in computer vision with wide-ranging applications in surveillance, traffic monitoring, anomaly detection, and human activity analysis. Traditional background subtraction techniques based on Gaussian Mixture Models (GMM) often suffer from limitations in dynamic environments, such as illumination changes, shadows, and complex backgrounds. This paper proposes a hybrid motion detection approach that integrates a Modified Gaussian Mixture Model (GMM) with a K-Nearest Neighbor (KNN)-based background subtraction method to improve robustness and precision. The system models each pixel as a mixture of Gaussians for statistical adaptability, while leveraging KNN's non-parametric classification to handle rapidly changing scenes. Experiments were conducted on multiple video datasets under diverse conditions, including normal lighting, shadow interference, night scenes, and adverse weather. Comparative evaluations against conventional GMM and standalone KNN approaches demonstrate that the proposed hybrid model achieves higher accuracy (95.2%), improved precision (74.1%), and superior F1-scores (68.7%), with significant reductions in false positives and false negatives. These results confirm that the integration of parametric and non-parametric methods enhances motion detection reliability in real-time applications. The proposed framework offers a scalable and efficient solution for intelligent video surveillance and related vision-based monitoring systems.

Key Words: *Motion detection, Background Subtraction Technique, Object recognition, Video surveillance, Gaussian Mixture Model(GMM), Contour Analysis.*

Date of Submission: 14-09-2025 Date of Acceptance: 24-09-2025

I. Introduction & Literature Review

The increasing need for automated detection systems across personal, commercial, industrial, and military domains has spurred the development of Video Analytics, promising to simplify lives and equip us to stay competitive amidst evolving technologies [1]. However, this trend also necessitates an examination of the challenges inherent in automated video surveillance scenarios. Despite humans' remarkable decision-making abilities, sustaining concentration levels proves challenging, with studies indicating that up to 90% of displayed information is overlooked after just 20 minutes of monitoring [5]. As Closed-Circuit Television (CCTV) systems proliferate, individuals may find themselves tasked with continuously monitoring feeds from numerous cameras around the clock. This underscores the necessity for automatic systems capable of analyzing and archiving video footage from multiple cameras and sensors, seamlessly detecting ongoing events and facilitating data navigation through sophisticated user interfaces—a concept commonly referred to as Video Analytics [3, 7].

Recent advancements in computer vision underscore a heightened focus on developing systems for monitoring and detecting humanactivity [8]. Such innovations hold significant promise across personal, industrial, commercial, and military spheres, empowering us to stay abreast of emerging technologies and tackle the complexities inherent in automated video surveillance. Video surveillance endeavors to detect, classify, and track objects across image sequences, aiding human operators in understanding and characterizing object behavior. These systems play a critical role in monitoring sensitive areas such as airports, banks, parking lots, and national borders.

The operational framework of automated video surveillance systems encompasses stages such as object detection, classification, andtracking, with motion detection serving as a foundational step. Motion detection aims to segment regions of interest corresponding to moving objects from the background, laying the groundwork for subsequent processes. Various techniques including frame differencing, adaptive median filtering, and background subtraction are employed to extract objects from stationary backgrounds, with the latter being the most widely utilized approach. Background subtraction involves crucial steps such as background modeling and foreground detection; wherein statistical descriptions of the background scene are used as reference frames to extract objects fromvideo frames. However, in scenarios featuring quasi-stationary backgrounds, such as swaying trees, flags, or water bodies, accurately detecting moving objects poses challenges. In such instances, traditional

single-model background frames may prove insufficient, necessitating the adoption of adaptive background modeling techniques for precise object detection amidst dynamic backgrounds [9,14]. Motion-based object detection is a fundamental approach in computer vision and video analysis that focuses on identifying and localizing objects in a scene based on their movement characteristics. This technique leverages the concept that moving objects stand out against a relatively static background, making motion a key indicator for detecting and tracking objects [2]. Researchers have proposed many methods, and background subtraction has gradually become the dominant solution for moving object detection. There are three main types of methods for moving object detection, Background Subtraction, Frame Differencing and Optical Flow. The first method, Background Subtraction such the identification of moving individuals within video footage captured by stationary cameras is frequently accomplished through background subtraction techniques. This methodology revolves around isolating moving objects by analyzing the contrast between t he current frame and a reference frame known as the "background frame" [4]. The background image serves as a static representation of the scene devoid of any dynamic elements and requires regular updates to accommodate changes in lighting and spatial configurations. Advanced models have expanded the scope of "background subtraction" beyond its literal interpretation. Background subtraction is the prevailing technique for motion detection, leveraging disparities between the present image and a stored background image to pinpoint areas of motion and providing pertinent object-related data [6]. The efficacy of this method hinges upon the accurate initialization and continuous updating of the background image, both of which profoundly influence theprecision of test outcomes. Consequently, this study employs a robust approach to initialize and dynamically update the background image.

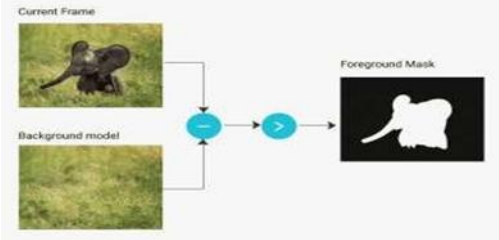

Fig.1 Foreground mask generated after background subtraction.

The second one is Frame differencing which involves determining the presence of moving objects by computing the disparity between two consecutive images [10]. Its computational simplicity and ease of implementation make it suitable for a wide range of dynamic environments. However, it often fails to provide comprehensive information about moving objects, leading tophenomena such as empty frames and consequently reducing the accuracy of object detection. The method, Optical flow refers to the visual pattern of object motion in a scene resulting from the relative movement between the scene and a camera (or observer). The optical flow method is employed in various applications such as enhancing video compression, segmenting images for object tracking, and estimating motion vectors for moving objects [11]. In motion detection using optical flow, the approach involves calculating the optical flow field within an image or video frame. This field represents the displacement of pixels over time, allowing the detection of moving objects based on the flow vectors. Despite its effectiveness, optical flow methods are typically

computationally intensive and time-consuming. Optical flow techniques can stabilize the background plane by modeling background motion, making it feasible to detect motion even in scenarios involving a moving camera and dynamic background. However, many optical flow methods require significant computational resources and are not suitable for real-time applications without specialized hardware[12].

II. Background Subtraction Algorithm

A fundamental stage in the background subtraction (BS) process involves background modeling, a crucial step wherein a fixed image is chosen to serve as the backdrop, excluding all moving elements.

Fig.2 Foreground Mask (Ft) after background subtraction,

The absolute difference (D_t) between the current frame (I_t) and the static image is computed for every frame in the video sequence. This static image, often referred to as the reference background or background model, is denoted as B_t-1. Consequently, the disparity image (D_t) is obtained through the following equation:

$D_t = |I_t — B_t—1|$ (1)

A foreground mask (Ft) is generated through the application of a threshold to a different image.

$$F_t = \begin{cases} 1, D_t > Th \\ 0, else \end{cases} \quad (2)$$

Illustrated in Fig.2 is an instance of the background subtraction procedure. To accommodate alterations within the scene over time, ensuring adaptability of the background model is imperative. The algorithm governing this learning phase necessitates gradual andonline execution. Employing a straightforward recursive technique, all pixels are updated utilizing an Infinite Impulse Response (IIR) filter.

$B_t = (1 — \alpha)\ B_t—1 + \alpha I_t$ (3)

where α is learning rate which is a constant in the range [0, 1]. $B_t—1$ and I_t are background and current image at time t-1 and t, respectively. ($I_t\ ^B t\text{-}1\ ^F t$).

Background subtraction algorithms can be mainly divided into three groups of learning: Unsupervised background subtraction, supervised background subtraction based on deep learning and semi-supervised background subtraction. Unsupervised methods focus on studying the internal mechanism of foreground detection and segmentation using a robust model that adapts to complex situations to represent it and constantly updates the model over time. The accuracy of a method is related to the strategy for establishing the background model, as well as the updating strategy. Clustering is a fundamental concept in unsupervised machine learning, where the goal is to group similar data points based on their inherent characteristics, without the need for labeled training data. Two popular clustering algorithms: K-Means and Gaussian Mixture Models (GMM).

Background Modeling Using K-Means

Background modeling using k-means clustering is a technique employed in computer vision to separate foreground objects from thebackground in video sequences. The process involves representing each pixel as a data point and applying k-means clustering togroup similar pixels into clusters [15]. The cluster representing the most frequently occurring pixel values across frames is consideredas the background model. This model is then used to identify and extract foreground objects by subtracting the background from eachframe. K-means clustering offers a straightforward and computationally efficient approach to segmenting pixels based on their characteristics, making it suitable for real-time applications such as surveillance, motion detection, and video analytics[16]. However, challenges such as handling dynamic backgrounds and selecting appropriate parameters for clustering require careful considerationto ensure accurate and robust background modeling.

Background Modeling Using GMM

Gaussian Mixture Models (GMMs) are employed for background modeling in video sequences by representing pixel intensities as a blend of Gaussian distributions, with each distribution representing a potential background appearance. Throughout the modeling process, GMMs dynamically learn and update the statistical properties of the background by adjusting the parameters (mean and covariance) of the Gaussian components over time [24]. This adaptive learning allows the model to adapt to gradual changes in the background scene. When processing a new frame, pixels that substantially diverge from the learned background model in terms of intensity distribution are classified as foreground, indicating the presence of moving objects or anomalies. GMM-based background modeling is effective for detecting and segmenting moving objects in video surveillance and related applications [27], leveraging the probabilistic nature of GMMs to capture complex background variations and foreground dynamics [17].

A: Building the Mathematical Model of the GMM*:* In the mathematical framework of Gaussian Mixture Models (GMM), several key components are defined to facilitate the understanding and estimation of complex data distributions.

The Probability Density Function (PDF) of a Gaussian distribution*:* serves as a fundamental building block. It encapsulates the likelihood of observing a specific data point within the distribution parameters such as the mean (μ) and variance (σ^2) characterize this PDF Fig. 3, influencing the shape and spread of the distribution.To get the PDF of the GMM we will start with the exponential function and try to manipulate and modify it until we get the multivariate Gaussian component which is the building block of the GMM. Where e^x increases rapidly on the positive side, while e^{-x} decreases rapidly on the negative side. The squared input value e^{x2} results in two characteristics: it imparts symmetry to the function and accelerates the growth of the function's output even more rapidly. On the other hand, the reversedsquared input value e^{-x2} where we get the familiar bell-shaped curve,

however, this does not conform to the normal distribution, and several additional elements must be incorporated to achieve it which are:

- The Mean μ (we want to change the center) we can do this by shifting the x-axis by a value equal to the μ ($e^{-(x-u)2}$).
- Variance σ (we want to control the width) we can do this by controlling the strength of the input value by multiplying itwith a constant known as standard deviation or sigma $e^{-(\sigma x-u)2}$

Mixture model: GMM represents the background scene as a composite of multiple Gaussian distributions, wheree adistribution, or component, is defined by its mean vector and covariance matrix. These components are weighed by mixing coefficients that determine their relative contributions to the overall background model [23]. The GMM is a probabilistic model that captures the underlying variability and complexity of the background by blending different Gaussian patterns Fig. 3. This mixture modeling approach is commonly used in tasks like background subtraction and anomaly detection in computer vision and image processing applications [25], providing a versatile framework for modeling and understanding diverse background scenes based on observed data distributions.

B: Initializing the GMM Parameters: The initialization of parameters for GMM based background modeling involves several key decisions. First, the number of Gaussian components (K) needs to be determined, reflecting the complexity of the background scene. This value is typically selected based on empirical analysis of the scene's variability and the desired granularity of the model. Secondly, the learning rate (α) dictates the speed of adaptation to changes in the scene; a smaller αresults in slower updates, allowing for more stable background modeling, whereas a larger α facilitates quicker adaptation to dynamic scenes. The threshold (τ) is crucial for pixel classification, where pixels are categorized as background or foreground based on their likelihoods under the learned background model; choosing an appropriate threshold is vital for accurate foreground detection. Lastly, selecting the covariance type (e.g., diagonal or full covariance matrices) for each Gaussian component impacts the flexibility and complexity of the model in capturing the background's statistical characteristics [26]. These initialization choices play a critical role in the effectiveness and performance of GMM-based background modeling algorithms, influencing how well the model adapts to scene variations and distinguishes moving foreground objects from the static background [18,19,20].

C: Modeling Pixels: GMM assumes that the observed data is generated from a mixture of multiple Gaussian distributions, each with its own set of parameters (mean and covariance matrix) and a weight representing its contribution to the overall distribution.

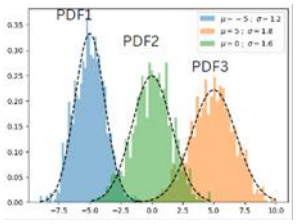

Fig. 3 Probability Density Function (PDF) of a Gaussian distribution

Initialization*:* Initialize the Gaussian mixture parameters (mean, covariance, and weight) for each component.
Estimation*:* Update the parameters iteratively as new observations are made. This involves updating the mean, covariance, and weight ofeach component based on the current pixel value and the learning rate.
Likelihood Calculation*:* the goal is to compute the probability or the likelihood of the observed pixel value given or each Gaussian components of the model. Each pixel inthe current frame is evaluated against each Gaussian component (distribution) to determine how likely it is that the pixel belongs to each distribution. For a pixel value x, and a Gaussian component j with mean μ_j, covariance matrix Σ_j, and weight ω_j, the likelihood P (x | j) is given by the Gaussian probability density function (PDF):

$$P(X|j) = \frac{1}{(2\pi)^{\frac{d}{2}}(\sum_j)^{\frac{1}{2}}} exp(-\frac{1}{2}(X-\mu_j)^T \sum_j^{-1}(X-\mu_j)) \qquad (4)$$

where d is the dimensionality of the feature vector (typically, d = 1 d=1 for grayscale pixel values and d = 3 for RGBcolor values) [30]. The overall likelihood of the pixel value under the mixture model is the weighted sum of the likelihoods for each Gaussiancomponent:

$$p(\chi) = \sum_{j-1}^{k} w_j P(\chi|j) \qquad (5)$$

where K is the number of Gaussian components.
Background Modeling*:* Classify the pixel as background if its likelihood under the background model exceeds a predefined threshold. Otherwise,consider it as part of the foreground.
History: The history parameter determines the number of past frames (history) that are used to initialize this background model.
1. Initialization and Accumulation of History
2. If the history buffer is not yet full, new frames are added to it.
3. Else If the history buffer is full, the oldest frame is removed, and the new frame is added.

Background_history$[i] = frame$

Background Model Update:
1. Start with the first frame in history as the initial background model.
2. Update the background model by blending each historical frame with the current background model using the learning rate α.

Background_model = $\alpha . hist\_frame + (1 - \alpha) . background\_model$
Background_model = background_history$[0]$

III. Proposed System (Modified GMM& KNN-Based Background Subtraction)

Foundation

In the realm of traffic surveillance systems, Friedman and Russell [1,7] introduced a method to characterize background pixels by employing a blend of three Gaussian distributions representing distinct elements: the road, vehicles, and shadows. They initialized this model using an Expectation-Maximization (EM) technique [6]. Subsequently, they manually assignedlabels to these Gaussians based on a heuristic approach: the darkest Gaussian component was designated as representing shadows, while the remaining two components were identified as representing the road and vehicles, respectively, with the vehicle componentdetermined by the largest variance. This configuration remained constant throughout the system's operation. Each pixel underwent comparison with these Gaussians to determine foreground detection, with classification based on the closest matching Gaussian. Maintenance of this model was achieved through the utilization of an incremental EM technique to enable real-time adaptation [1,7]. Stauffer and Grimson [1] further developed this concept by incorporating recent color property variations of each pixel into their Gaussian model.

Pixel processes and parameters initialization

At any time t, we have a history of a particular pixel (X0, Y0). Models the values of particular pixels as a mixture of Gaussians. At any time, t, we have k distributions of Gaussians for each pixel (used 3 Gaussians). For each Gaussian, we have:

$\boldsymbol{\omega i, t}$: is an estimated weight of ith Gaussian in the mixture, at time t. initialized with uniform distribution which led to:
1. Equal Starting Point: By initializing the weights uniformly, each Gaussian component starts with an equal probability of representing the pixel's value. This prevents any initial bias towards specific components and ensures that all components are considered equally likely to begin with. Over time, the weights will adapt based on the observed pixel values, allowing the model to accuratelyrepresent the pixel distribution.
2. Stability in Early Iterations: Uniform initialization provides stability in the early stages of the algorithm. Since no prior information is available aboutwhich Gaussian might be more relevant, giving each component an equal weight helps in avoiding any premature convergence to incorrect components. This can help in reducing the initial noise and variability in the background model.
3. Simplified Implementation: Using a uniform distribution simplifies the implementation and initialization process. It avoids the need for complexcalculations or heuristics to determine initial weights, making the algorithm easier to implement and understand.
4. Fair Competition Among Components: When the weights are initialized uniformly, each Gaussian component has an equal chance of representing the observed pixel values. This fair competition ensures that the most appropriate Gaussian components will emerge over time as theycompete based on their fit to the data. The components that best match the observed values will naturally receive higher weights, leading to a more accurate and robust background model.
5. Improved Convergence: Uniform initialization can lead to improved convergence properties. Since all components start with equal importance, thealgorithm is less likely to get stuck in local minima where certain components dominate prematurely. This helps in achieving a more accurate and stable background model as the algorithm progresses.

$\mu_{i,t}$: is the mean value of i[th] Gaussian in the mixture, at time t, initialized with vector of zeros which lead to:

1. The model start with a simple assumption and quickly adapts to the actual data as it processes more frames.
2. Handling Unknown Initial Conditions: When there is little to no prior knowledge about the data distribution, initializing with zeros is a reasonable choice. It allows the algorithm to explore the data and converge to the appropriate means based on the observed data rather than being influenced by potentially incorrect initial guesses.

$\Sigma_{i,t}$: covariance matrix of i[th] Gaussian in the mixture, at time t: This ensures that the initial covariance is equal in all directions and prevents singularities. A singular covariance matrix leads to degenerate Gaussian distributions, affecting the model's ability to represent data distributions accurately. By initializing covariance matrices with non-zero values, such as small positive values or an identity matrix, we can prevent singularities from occurring. This initialization ensures that the matrix remainsinvertible and avoids numerical issues during computations. For computational reasons, we assumed that the RGB color components are independent and have the same variances. So, the covariance matrix is of the form:

$$\sum_{i,t} = \sigma_{i,t}^2 I \quad (6)$$

Now, the probability of observing the current pixel value is:

$$P(X_t) = \sum_{i=1}^{K} \omega_{i,t}.\eta(X_t, \mu_{i,t}, \Sigma_{i,t}) \quad (7)$$

Where 'K' is the number of Gaussian distribution, ω, μ, ∑ as stated previously, and η is the gaussian probability density function Eq. (8).

$$\eta(X_t,\mu,\Sigma) = \frac{1}{(2\pi)^{\frac{n}{2}}|\Sigma|^{\frac{1}{2}}} exp(-\frac{1}{2}(X_t-\mu)^{\Sigma^{-1}(X_t-\mu)}) \quad (8)$$

So, each pixel is characterized by a mixture of K Gaussians. Once the background model is defined, the different parameters of the mixture of Gaussians must be initialized. The parameters of the mixture of Gaussians model are the number of Gaussians K, the weight $\omega_{i,t}$, t associated to the i^{th} Gaussian at time t, the mean $\mu_{i,t}$ and the covariance matrix $\Sigma_{i,t}$.

Background model estimation*:* The Gaussians are ordered by the value of ω/σ Then, the first B distributions Eq. (9) are chosen as the background model [28],

$$B = argmin_b(\sum_{i=1}^{b} \omega_{i,t} > T) \quad (9)$$

where T is a measure of the minimum portion of the data that should be accounted for by the background [29].

Update the mixture model: Every new pixel value, X_t, is checked against the existing K Gaussian distributions until a match is found. A pixel matches aGaussian distribution if the Mahala Nobis distance Eq. (10).

$$\text{sqrt}((X_{t+1}-\mu_{i,t})^T.\sum_{i,t}^{-1}(X_{t+1}-\mu_{i,t})) < k\sigma_{i,t} \quad (10)$$

Where K is a constant threshold equal to 2.5, now two cases can occur:

Case 1: A match is found with one of the K Gaussians. In this case, if the Gaussian distribution is identified as a background one,the pixel is classified as background else the pixel is classified as foreground. For the matched component, the update is done as follows:

$$\omega_{i,t\square 1} \square (1\square\alpha)\,\omega_{i,t} \square\alpha \quad (11)$$

where α is a constant learning rate.

$$\mu_{i,t\square 1} \square(1\square\rho)\,\mu_{i,t} \square \rho.X_{t\square 1} \quad (12)$$

$$\sigma^2_{i,t\square 1} \square (1\square\rho)\,\sigma^2_{i,t} \square \rho(X_{t\square 1} \square \mu_{i,t\square 1}).(X_{t\square 1} \square\mu_{i,t\square 1})^{\mathrm{T}} \quad (13)$$

where $\rho \square \alpha.\eta\ (X_{t\square 1}, \mu_i, \square_i)$

For the unmatched components, μ and $\square$ are unchanged, only the weight is replaced by

$$\omega_{j,t\square 1} \square (1\square\alpha)\,\omega_{j,t} \quad (14)$$

Case 2: No match is found with any of the K Gaussians. In this case, the pixel is classified as foreground & the least probabledistribution k is replaced with a new one with parameters:

$$\omega_{k,t\square 1} \square \text{Low Prior Weight} \quad (15)$$

$$\mu_{k,t\square 1} \square X_t \square 1 \quad (16)$$

$$\sigma^2_{k,t\square 1} \square \text{ Large Initial Variance} \quad (17)$$

Once the maintenance of parameters is made, foreground detection can be made, and so on. Complete studies on the signification andthe setting of the parameters can be found in [22, 23].

Learning rate:

The alpha parameter in the algorithm has a significant effect on its performance, particularly in terms of how

quickly the background model adapts to changes in the scene. Here's how it works:

- Low Alpha: When alpha is low, the background model changes slowly over time. This means that the algorithm is more resistant to short-term fluctuations or noise in the video frames. However, it may take longer for the background model to adapt to significant changes in the scene, such as sudden lighting changes or moving objects.
- High Alpha: Conversely, when alpha is high, the background model adapts quickly to changes in the scene. This can help thealgorithm respond rapidly to changes like moving objects or changes in lighting conditions. However, it may also make the algorithm more sensitive to noise, leading to false detections or instability in the background model.

KNN-Model

The KNN model is implemented through two main steps: **Background Decision**: This step involves classifying each pixel as either background or foreground by comparing it to itshistorical pixels, **History Update**: This step updates all historical pixels at varying rates [31].

Background Decision: For each pixel, historical values from previous frames are stored in three different history lists: short, medium, and long as shown in Fig. 4. Each history list contains a predefined number of samples, such as N=7, where each sample is represented by a structure storing the red,green, and blue channel values (Ri, Gi, and Bi) for the i-th sample pixel, xi. Additionally, a flag (Flagi) is used to indicate if xi is apotential background pixel (Flagi = 1) or a foreground pixel (Flagi = 0). A potential background pixel is either classified as background or likely to be classified as such. When a new pixel, xt, is obtained from the next frame, the algorithm computes the Euclidean distance Dis(t,i) in RGB color spacebetween xt and each historical sample xi (i = 1, ..., 3N) in the three history lists.

$$\text{Dis(t,i)} = \sqrt{(Rt - Ri)^2 + (Gt - Gi)^2 + (Bt - Bi)^2} \qquad (18)$$

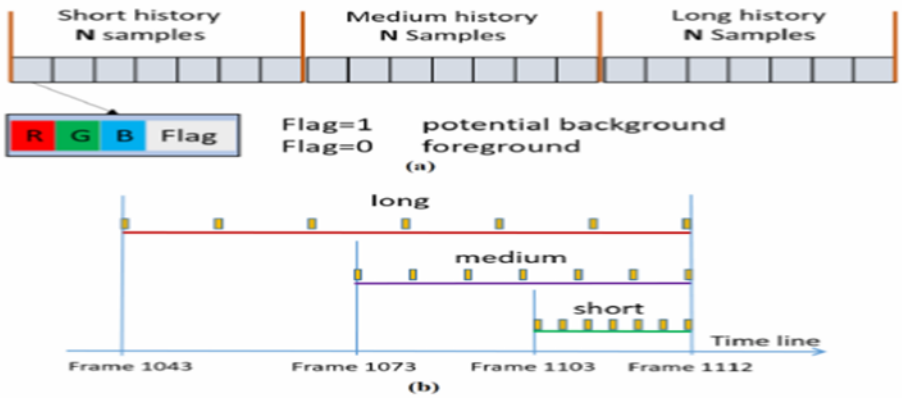

FIG. 4. The structure of the histories for a pixel in KNN background subtraction model. (a) Each of the three histories contains N samples composed of R, G, B, and a Flag value. (b) The three histories contain the same number of samples but cover different

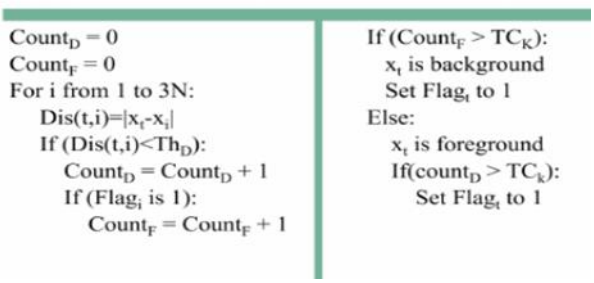

FIG. 5. Pseudo code for classifying a pixel x_t as background or foreground.

The process of classifying whether the new pixel x_t is background or foreground can be described algorithm in Fig.5. The algorithm iterates through all 3N historical samples, a counter (CountD) tracks the number of distances smaller than a threshold $Th\,D$. Simultaneously, another counter (CountF) counts the number of samples with Flag=1where Dis(t,i) < $Th\,D$. The values of CountD and CountF are then compared to a threshold $TC\ k$ to

classify x_t as either background or foreground[31]. The pixel x_t is immediately classified as background (value = 1) if there are more than $T_C\ k$ potential background pixels within the distance threshold T_hD . Otherwise, x_t is classified as foreground (value = 0). If CountD exceeds $T_C\,k$, x_t can be reclassified as potential background, setting its Flag value to 1. Each frame is processed in this way, resulting in a binary mask image where background pixels are labeled as 1 andmoving targets as 0.

History Update: The three history lists cover different time ranges, as shown in Fig. 4. For instance, the short history covers frames from 1103 to 1112, the medium history from 1073 to 1112, and the long history from 1043 to 1112.

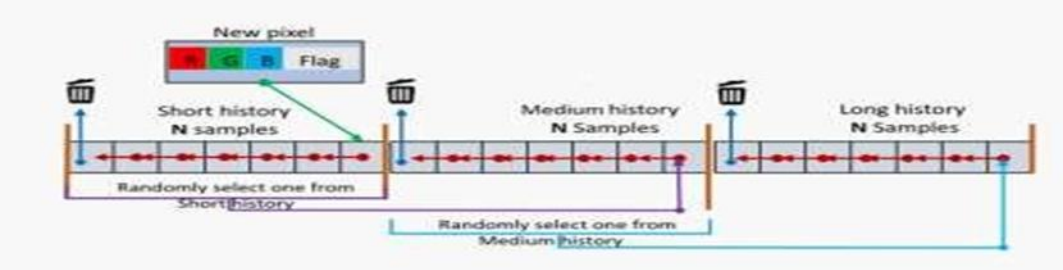

FIG. 6. History update.

This indicates that the short history contains only the most recent samples, while the medium and long histories include samples fromearlier frames with larger intervals between samples. Hence, the three histories are updated at different rates according to the followingEq. (19):

$$T_{short} = Round(\frac{log(0.7}{log(1-\alpha)} + 1$$

$$T_{mid} = Round(\frac{log(0.4)}{log(1-\alpha)} + 1 - T_{short}$$

$$T_{long} = Round(\frac{log(0.1)}{log(1-\alpha)} + 1 - T_{mid} - T_{short} \qquad (19)$$

Here, *Round(.)* is the rounding function, and α ($0 < \alpha < 1.0$) is the learning rate, set to 0.02 in this study. Once the time ranges are determined, the update ratio R_j is calculated as: $R_j=Round\ (T_j/N)+1$ Where T_j represents T_{short} , T_{mid} and T_{long} , and N is the number of samples in each history (e.g., N=7 in Fig. 4). Consequently,the short history is updated most frequently, while the long history is updated the least [31].

Fig. 6 details the updating process for the three histories. Samples in each history are managed by a queue, with the head on the leftand rear on the right. When a new pixel arrives, it is first classified as potential foreground (Flag = 1) or background (Flag = 0), then added to the rear of the short history.

Since the short history only retains N samples, the oldest sample on the left is removed fromthe queue. For updating the medium history, a random sample from the short history is appended to the medium history's rear, and theoldest sample is removed from the medium history. Similarly, the long history is updated by selecting a random pixel from the mediumhistory. This process ensures that the samples are sorted on time, and the earliest one is discarded after updating the short history. When the algorithm iterates from the first frame to the last, the three histories are updated at different rates.

By combining K-Nearest Neighbors (KNN) and Gaussian Mixture Model (GMM), a motion detection system can effectively utilizethe strengths of both non-parametric and parametric approaches. KNN's adaptability and flexibility enable it to rapidly respond to changes in the environment, capturing complex and dynamic patterns in the background. This is particularly useful in situations wherethe background is constantly changing or contains frequent movements. GMM, on the other hand, provides a structured and detailed background model by representing the background as a mixture of Gaussian distributions. This parametric approach excels in distinguishing between foreground and background elements, even in the presence of subtle changes such as lighting variations andshadows.

By integrating these two methods, the hybrid system achieves a balance between responsiveness and stability, significantlyenhancing the accuracy, efficiency, and robustness of motion detection. This makes the system highly effective across a wide range ofenvironments and conditions, such as in surveillance systems where it can accurately detect intruders in both stable and dynamic settings, in traffic monitoring where it can reliably identify moving vehicles and pedestrians, and in wildlife monitoring where it cantrack animals in rapidly changing natural habitats. The hybrid approach not only reduces false positives and negatives but also ensuresreal-time performance, making it a versatile and powerful solution for various motion detection applications.

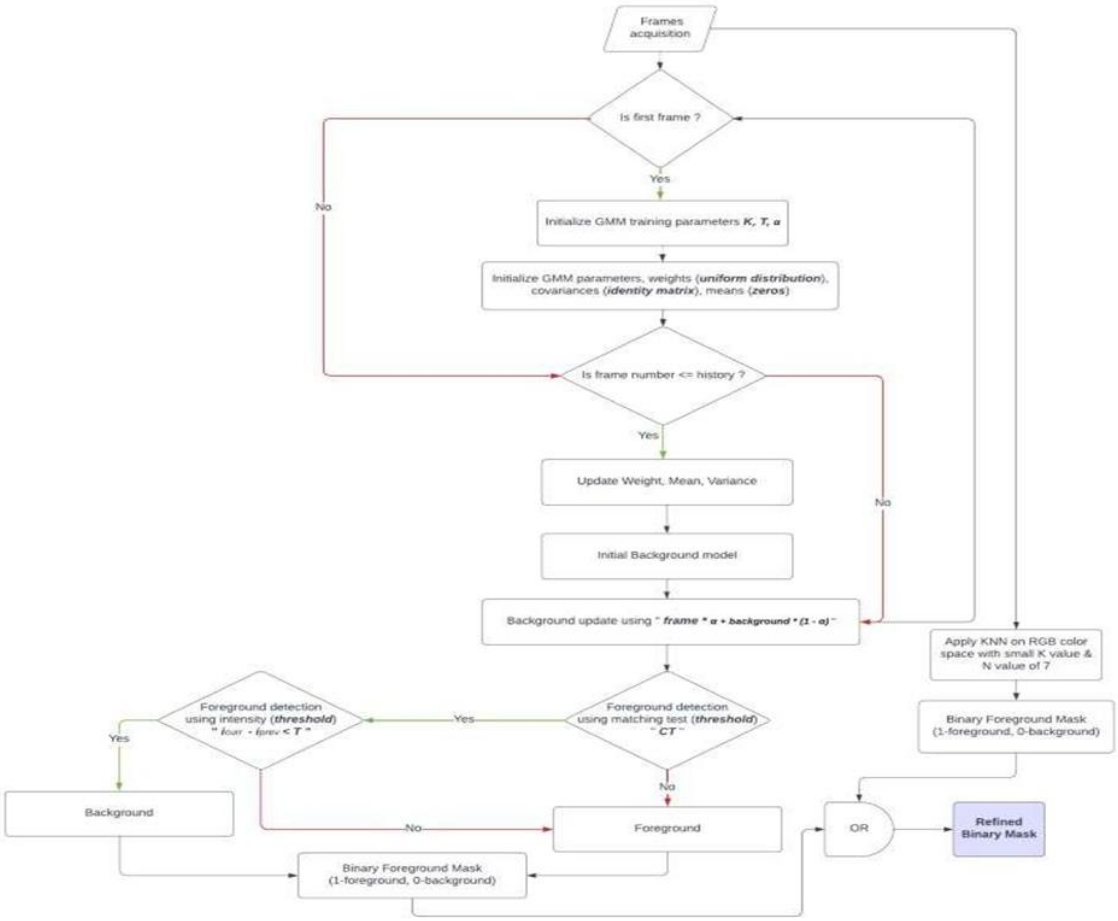

Fig. 7. Flow chart for hybrid-modified GMM-KNN based object detection.

IV. Testing & Evaluation Of GMM & Modified GMM

Experimental setup: A high fixed value for K was selected. Different combinations of α and T were evaluated across the dataset to determine the most effective pair for object detection across multiple videos. The Total Error (TE), a key performance metric, was introduced to identify the pair ofα and T with the lowest TE value, which represents the total misclassifications across all videos. The object detection system underwent testing with various α and T parameters across a range of background-covering sequences, with emphasis on sequences from the four datasets we used.

Training and testing: The dataset provider has outlined comprehensive recommendations for both training and testing procedures. Below are concisedescriptions of sequences, accompanied by instructions for training and testing.

Normal Case (NC): A girl with brown hair and wearing white & black clothes walks confidently across a green screenstudio. She smiles and makes eye contact with the viewer.

Shadow Case (SC): A girl in a long, flowing, floral clothes and a headscarf walking down a sidewalk. In the background is a large building with many windows. The girl's shadow appears throughout the video while she's walking.

1. **Night Case (NIC): A** man walking down a foggy street at night. The street is lit by lampposts that cast pools of lighton the sidewalk. The man is in the center of the frame,. He is wearing a long coat.
2. **Bad Weather Case (BWC): A** man walking down a sidewalk on a snowy street at night and the sky is dark and wearing a winter coat and hat, and appears to be trudging through the snow.

Results of GMM & Modified GMM: The performanceof the Gaussian Mixture Model (GMM) and our Modified GMM is evaluated through a series of metrics. Table 1 displays the coreperformance metrics for the GMM, specifically true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN). These metrics provide a foundational understanding of the model's ability to correctly identify motion versus background. Table

2, we further analyze the GMM performance through derived key metrics: accuracy, precision, recall, and F1 score, which offer a more holistic view of the model's effectiveness in various aspects. Tables 3 and 4 mirror the structure of Tables 1 and 2, respectively, but for the Modified GMM. This comparison allows us to evaluate the improvements brought by our modifications, demonstrating enhanced performance in motion detection through a detailed examination of both basic and derived metrics.

Table 1: Performance parameter evaluation of GMM, KNN and Hybrid Modified GMM & KNN (True Positive)

Dataset	True Positive		
	Basic GMM	**KNN**	**Modified GMM & KNN**
Normal Case (NC)	60441	34648	58947
Shadow Case (SC)	66329	17914	54095
Night Case (NIC)	11226	1666	10222
Bad Weather Case (BWC)	78486	12625	85013
Overall	**216,48**	**66,85**	**208,287**

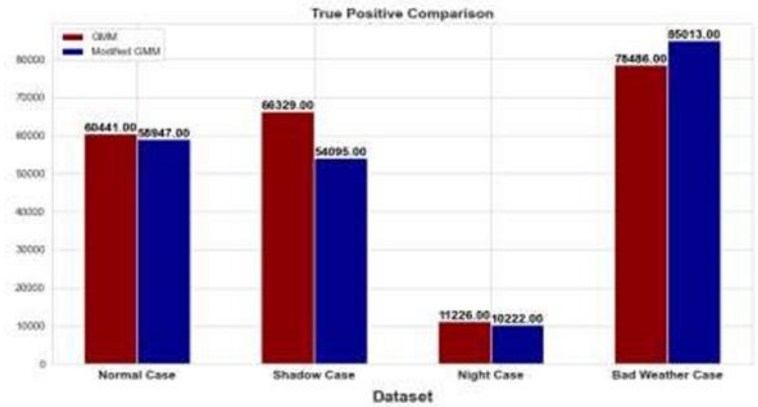

Fig. 6a. True positive of GMM compared to Modified GMM & KNN based object detection.

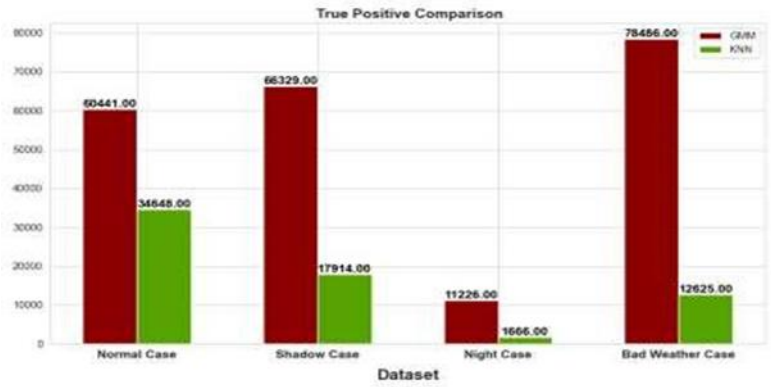

Fig. 6b. True positive of GMM compared to KNN based object detection.

Table 2: Performance parameter evaluation of GMM, KNN & Hybrid Modified GMM & KNN (True **Negative**)

Dataset	True Negative		
	Basic GMM	**KNN**	**Modified GMM & KNN**
Normal Case (NC)	1285445	1308352	1303511
Shadow Case (SC)	1129506	1172068	1126806
Night Case (NIC)	1351358	1367680	1362786
Bad Weather Case (BWC)	1269753	1274108	1269484
Overall	**5,036,062**	**5,122,208**	**5,062,587**

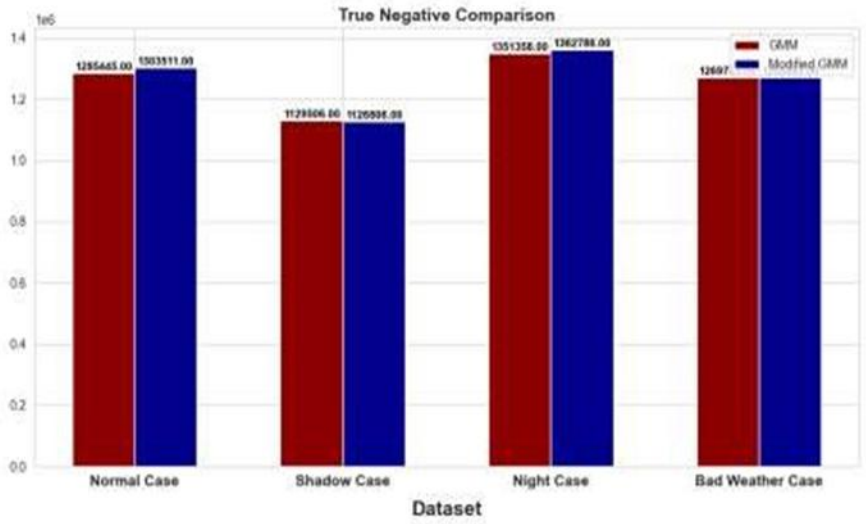

Fig. 6c. True Negative of GMM compared to Modified GMM & KNN based object detection

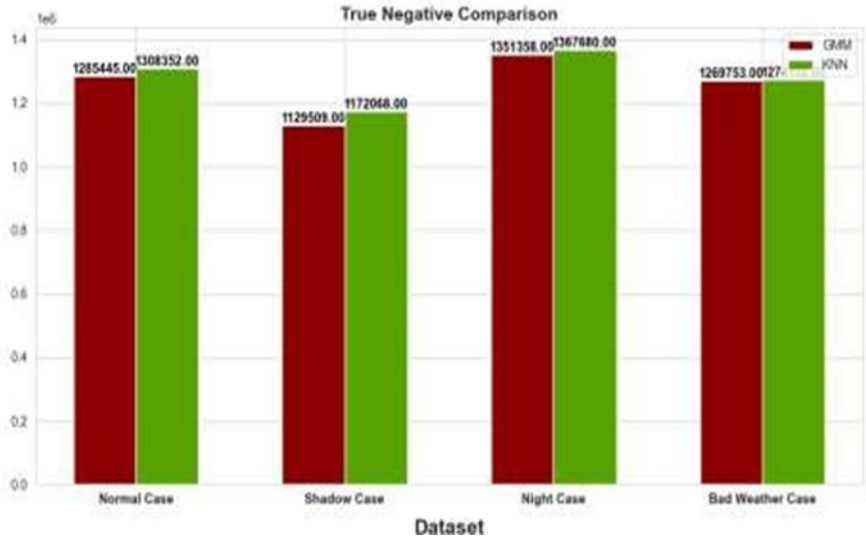

Fig. 6d. True Negative of GMM compared to KNN based object detection

Table 3: Performance parameter evaluation of GMM, KNN & Hybrid Modified GMM & KNN (False Positive)

Dataset	**False Positive**		
	Basic GMM	**KNN**	**Modified GMM & KNN**
Normal Case (NC)	23104	197	5038
Shadow Case (SC)	53688	11126	56388
Night Case (NIC)	16856	534	5428
Bad Weather Case (BWC)	8991	4636	9260
Overall	**102,64**	**16,49**	**76,11**

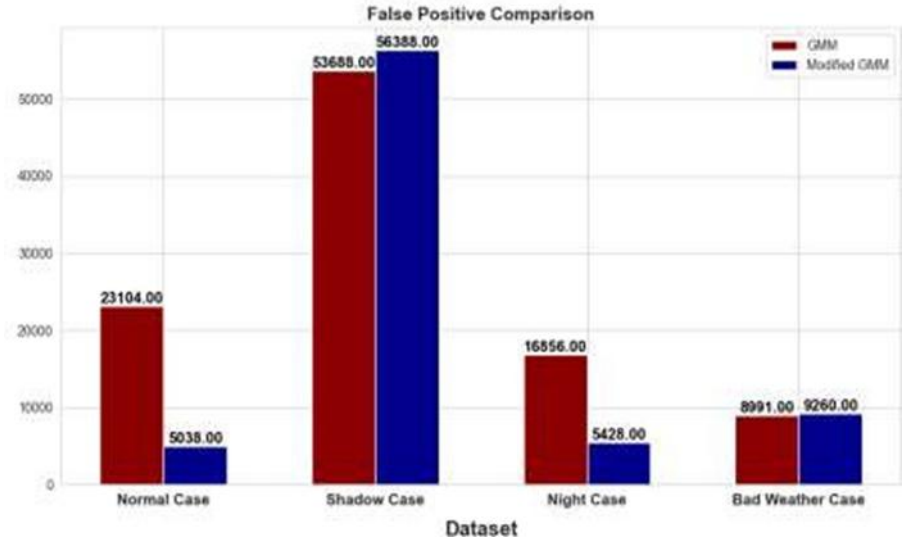

Fig. 6e. False Positive of GMM compared to Modified GMM & KNN based object detection.

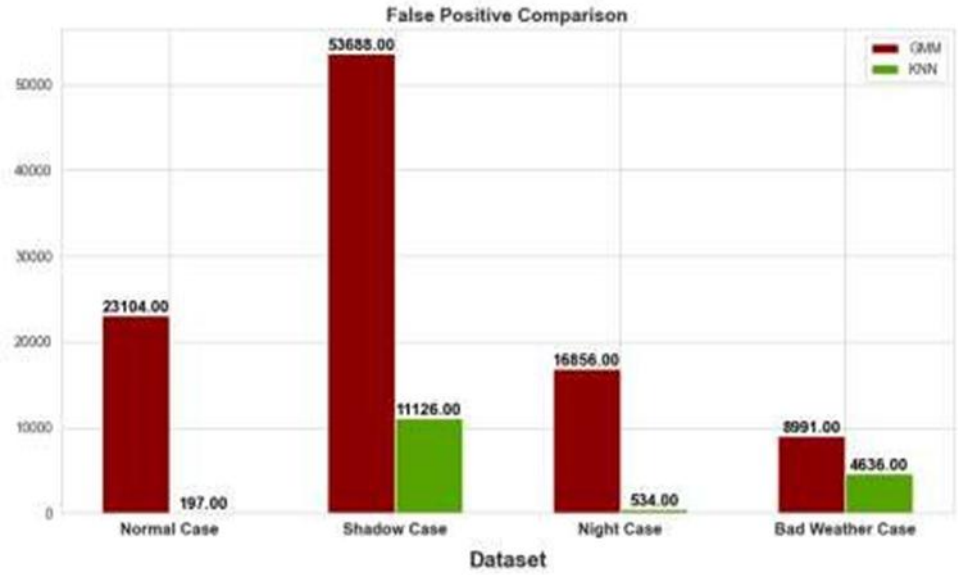

Fig. 6f. False Positive of GMM compared to KNN based object detection.

Table 4: Performance parameter evaluation of GMM, KNN & Hybrid Modified GMM & KNN (False Negative)

Dataset	**False Negative**		
	Basic GMM	**KNN**	Modified GMM & KNN
Normal Case (NC)	13410	39203	14904
Shadow Case (SC)	132877	181292	145111
Night Case (NIC)	2960	12520	3964
Bad Weather Case (BWC)	25170	91031	18643
Overall	174,42	324046	182,622

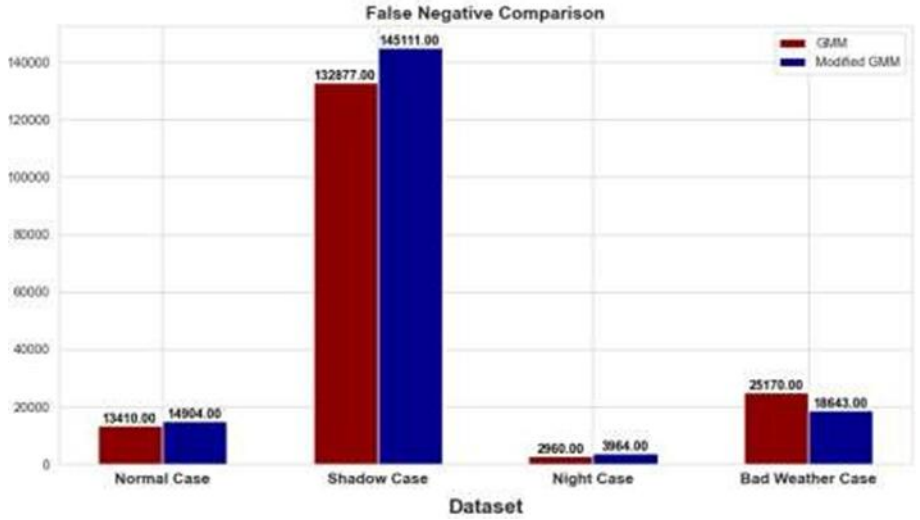

Fig. 6g. False Negative of GMM compared to Modified GMM & KNN based object detection.

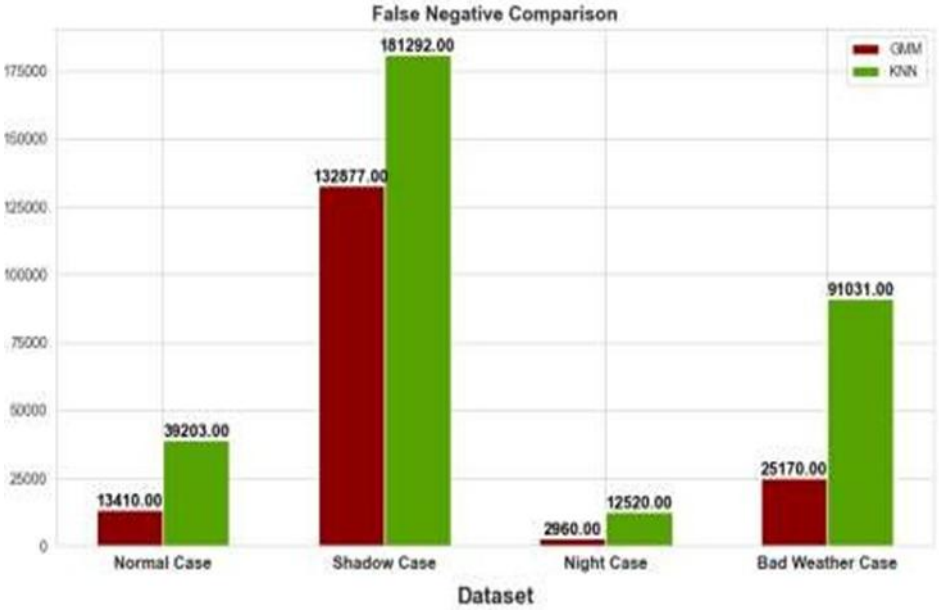

Fig. 6h. False Negative of GMM compared to KNN based object detection.

Table 5: Performance parameter evaluation of GMM, KNN & Hybrid Modified GMM & KNN (**Accuracy**)

Dataset	**Accuracy**		
	Basic GMM	**KNN**	**Modified GMM & KNN**
Normal Case (NC)	0.973	0.97	0.985
Shadow Case (SC)	0.865	0.86	0.854
Night Case (NIC)	0.98	0.99	0.99
Bad Weather Case (BWC)	0.975	0.93	0.979
Overall	94.82%	93.75%	95.2%

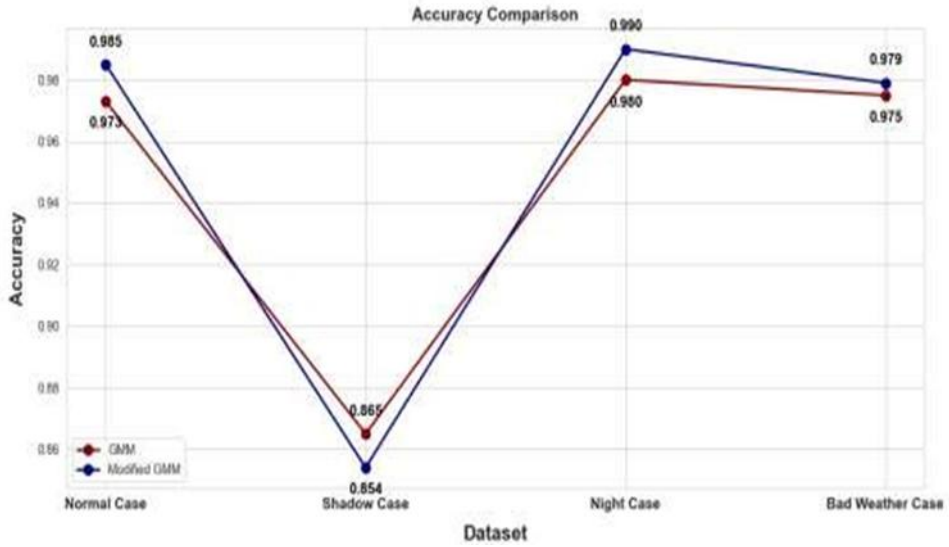

Fig. 6i. False Negative of GMM compared to Modified GMM & KNN based object detection.

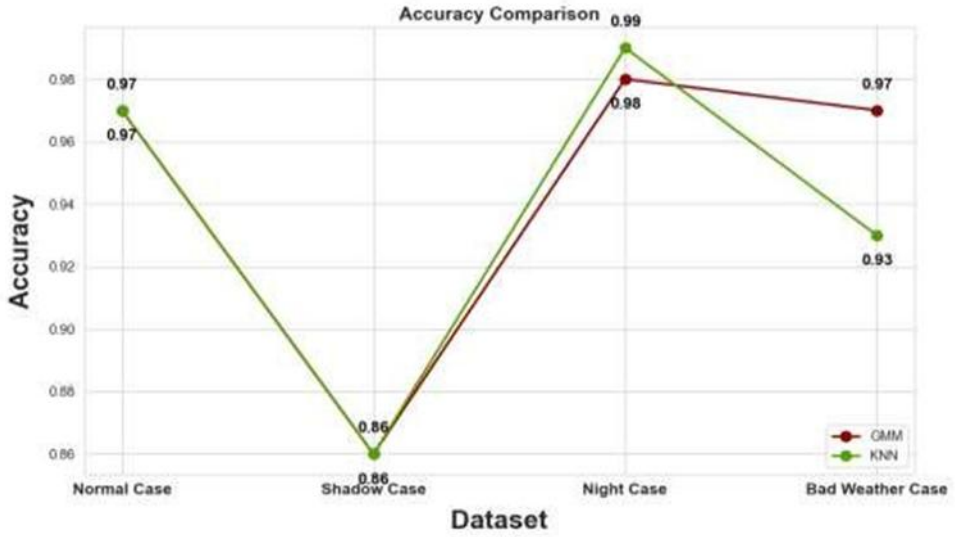

Fig. 6j. Accuracy of GMM compared to KNN based object detection.

Table 6: Performance parameter evaluation of GMM, KNN & Hybrid Modified GMM & KNN (Precision)

Dataset	**Precision**		
	Basic GMM	**KNN**	**Modified GMM & KNN**
Normal Case (NC)	0.723	0.99	0.921
Shadow Case (SC)	0.552	0.65	0.489
Night Case (NIC)	0.399	0.75	0.653
Bad Weather Case (BWC)	0.897	0.73	0.901
Overall	64.2%	78%	74.1%

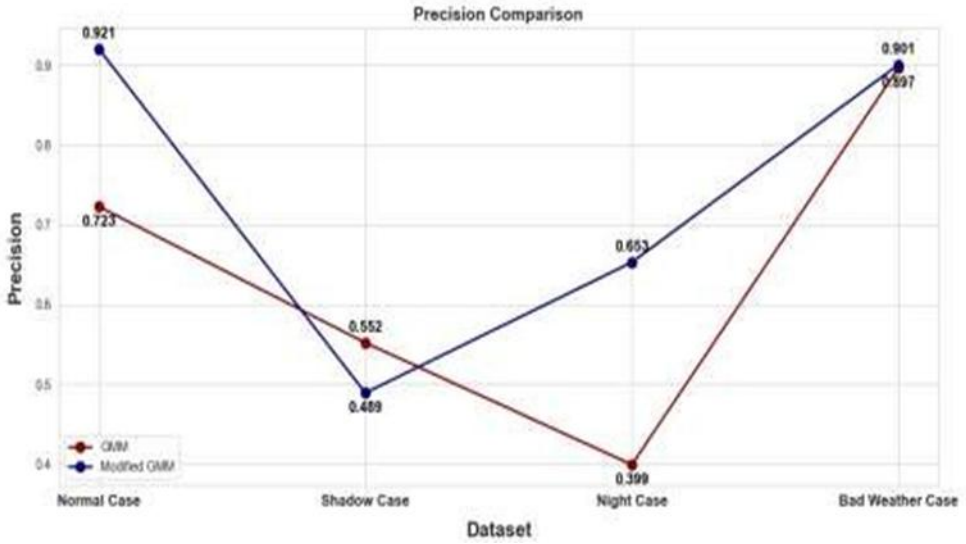

Fig. 6k. Precision of GMM compared to Modified GMM & KNN based object detection.

Table 7: Performance parameter evaluation of GMM, KNN & Hybrid Modified GMM & KNN (Recall)

Dataset	Recall		
	Basic GMM	**KNN**	**Modified GMM & KNN**
Normal Case (NC)	**0.818**	**0.46**	**0.798**
Shadow Case (SC)	**0.332**	**0.08**	**0.271**
Night Case (NIC)	**0.791**	**0.11**	**0.720**
Bad Weather Case (BWC)	**0.757**	**0.12**	**0.820**
Overall	**67.4%**	**19.25%**	**65.2%**

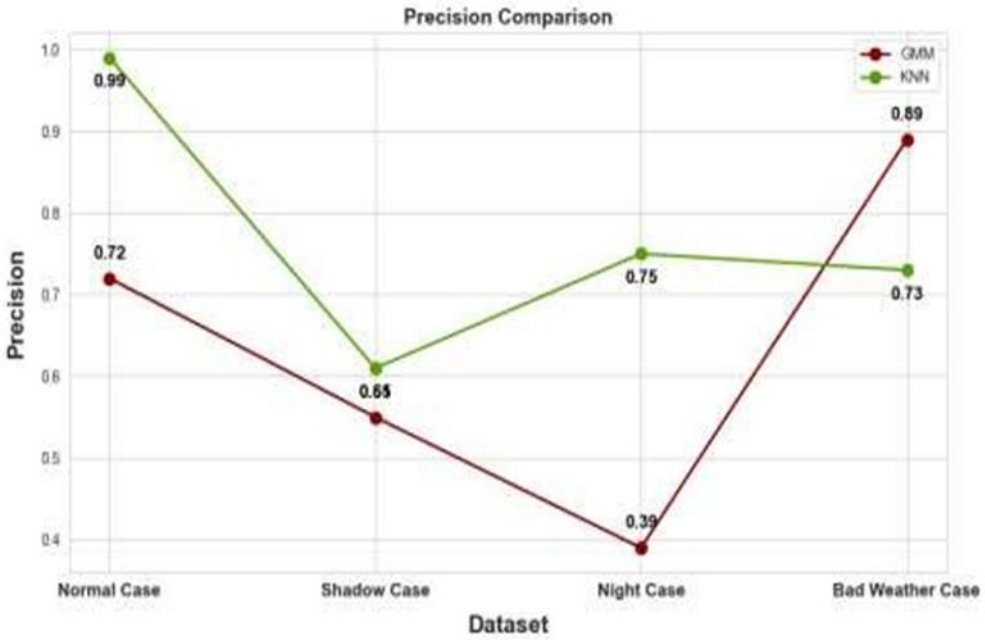

Fig. 6l. Precision of GMM compared to KNN based object detection.

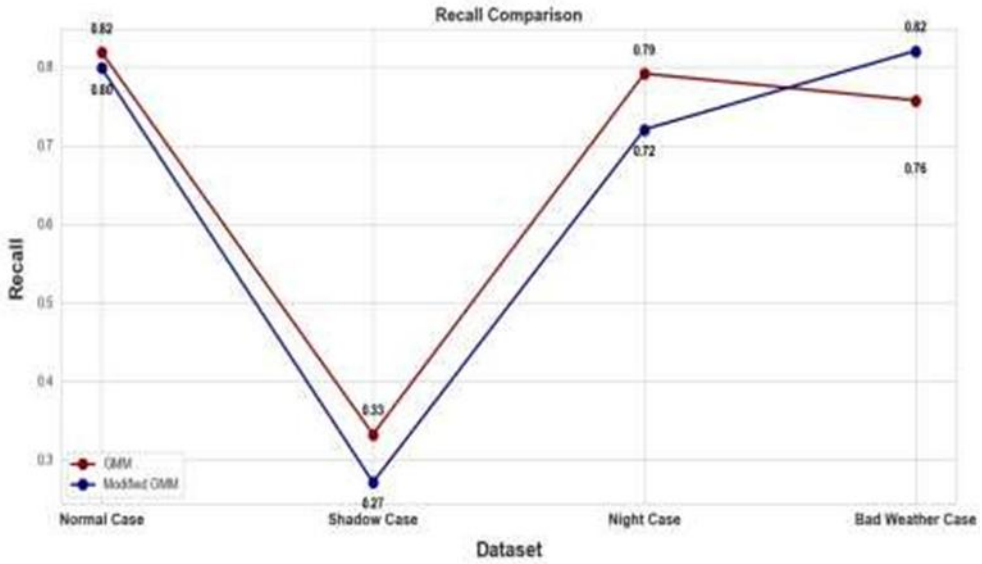

Fig. 6m. Precision of GMM compared to Modified GMM & KNN based object detection

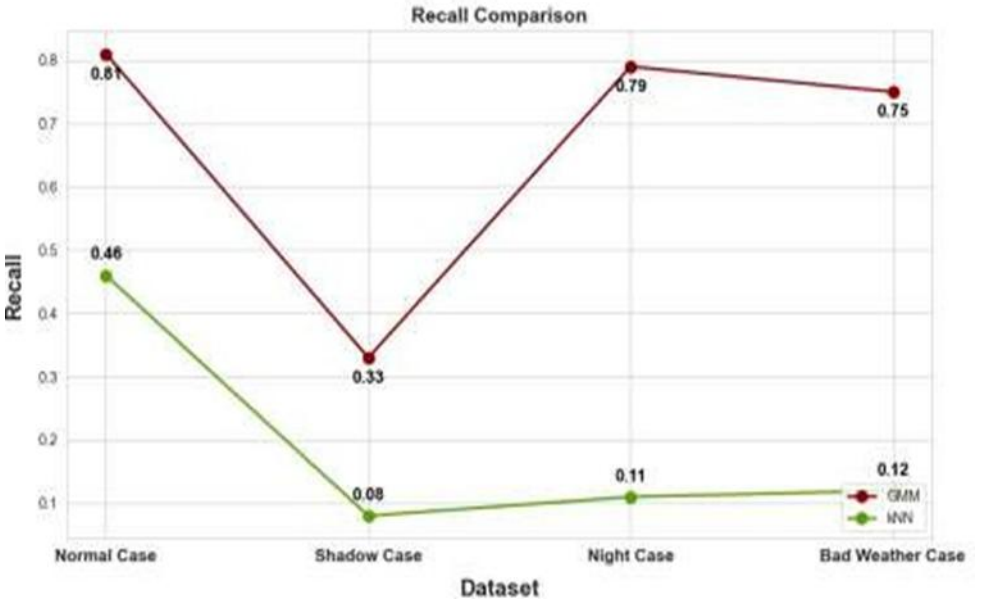

Fig. 6n. Precision of GMM compared to KNN based object detection.

Table 8: Performance parameter evaluation of GMM, KNN & Hybrid Modified GMM & KNN (F1 Score*)*

Dataset	**F1 Score**		
	Basic GMM	**KNN**	**Modified GMM &KNN**
Normal Case (NC)	0.768	0.46	0.855
Shadow Case (SC)	0.415	0.08	0.349
Night Case (NIC)	0.531	0.11	0.685
Bad Weather Case (BWC)	0.821	0.12	0.859
Overall	63.37%	19.25%	68.7%

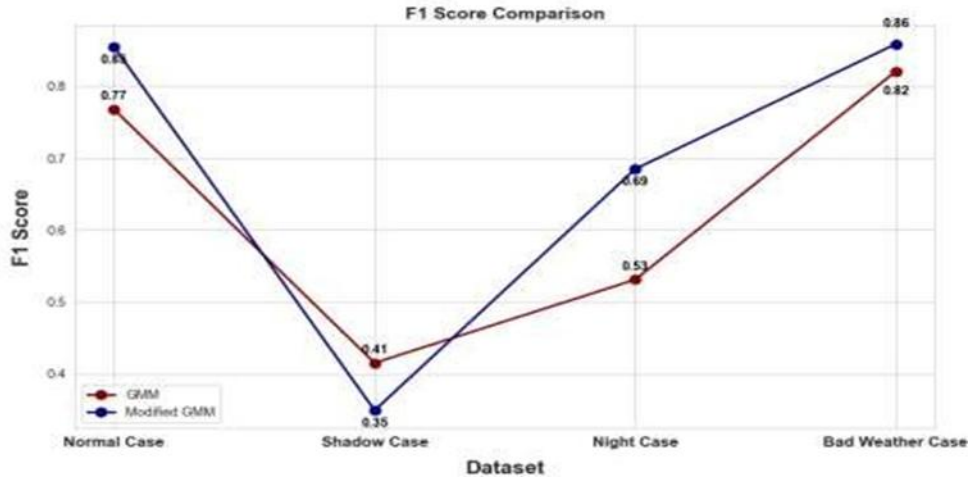

Fig. 6o. Precision of GMM compared to Modified GMM & KNN based object detection.

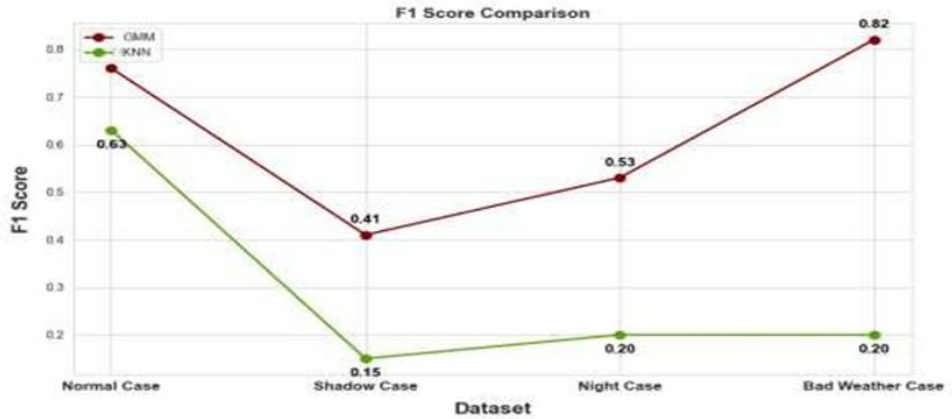

Fig. 6p. Precision of GMM compared to KNN based object detection.

Finally, the experimental evaluation across diverse datasets including normal lighting, shadow interference, night scenes, and adverse weather, demonstrated the superiority of the proposed hybrid Modified GMM and KNN approach. Compared to baseline GMM and standalone KNN methods, the hybrid system achieved higher overall accuracy (95.2%) shown at Table. 5 and their figures Fig, 6i, Fig. 6j, improved precision (74.1%) shown at Table. 6 and their figures Fig, 6k, Fig. 6l, and a stronger F1 score (68.7%) shown at Table. 85 and their figures Fig. 6o, Fig. 6p, while significantly reducing false positives and false negatives. These results confirm that integrating parametric and non-parametric approaches yields a more reliable framework for motion detection in complex environments. The performance improvements were consistent across all tested scenarios, underscoring the robustness and adaptability of the proposed model.

V. Conclusion And Comparison

This study introduced a hybrid motion detection framework that integrates a Modified Gaussian Mixture Model with K-Nearest Neighbor background subtraction, addressing key limitations of conventional approaches in dynamic and challenging environments. By combining the probabilistic rigor of GMM with the adaptability of KNN, the proposed system effectively balances stability and responsiveness, enabling accurate object detection under conditions such as illumination changes, shadows, occlusions, and dynamic backgrounds. The results highlight not only the enhanced detection performance but also the scalability of the method for real-time surveillance and monitoring applications. Importantly, this research contributes to advancing video analytics by demonstrating the benefits of hybrid ensemble models for robust motion detection. Future work will focus on optimizing computational efficiency, extending the framework to deep learning–based architectures, and exploring its applicability in broader domains such as autonomous navigation and intelligent transportation systems.

References

[1] Moving Object Detection Using Modified GMM Based Background Subtraction - Sciencedirect
[2] Yi G Ithan Dedeo G Lu," MOVING OBJECT DETECTION TRACKING AND CLASSIFICATION FORSMART VIDEO

SURVEILLANCE", August 2004.
[3] Paul Viola, Michael Jones, "Detecting Pedestrians Using Patterns Of Motion And Appearance", International Journal Of Computer Vision 63(2), 153-161, 2005.
[4] Neeti A. Ogale, "A Survey Of Techniques For Human Detection FromVideo,", Unpublished.
[5] [1302.1539] Image Segmentation In Video Sequences: A Probabilistic Approach (Arxiv.Org)
[6] Prithviraj Banerjee And Somnath Sengupta, "Human Motion Detection And Tracking For Video Surveillance,", Unpublished
[7] X. Niu, Y. Zhu, Q. Cao, X. Zhang, W. Xie, K. Zheng, An Online-Traffic-Prediction Based Route Finding Mechanism For Smart City, Int. J. Distributed Sens. Netw.11 (8) (2015), 970256.
[8] Bo Wu And Ram Nevatia, "Detection And Tracking Of Multiple Partially Occluded Humans By Baysian Combination Of Edgelet Based Part Detector" ,International Journal Of Computer Vision 75(2), 247-266, 2007
[9] Yang Ran, Issac Weiss, "Pedestrian Detecting Via Periodic Motion Analysis", International Journal Of Computer Vision 71(2), 143-160, 2007.
[10] Abhishek Kumar Chauhan, Prashant Krishan, Moving Object Tracking Using Gaussian Mixture Model And Optical Flow, Int. J. Adv. Res. Comput. Sci. Software Eng. (April 2013).
[11] Milin P Patel And Shankar K. Parmar, "Moving Object Detection With Moving Background Using Optic Flow," IEEE Conference On Recent Advances And Innovations In Engineering, Jaipur, Pp 1-6, 2014.
[12] Vaibhavi S Bharwad Et Al," Motion Detection For Intelligent Video Surveillance System: A Survey", 2015
[13] D. Reynolds, Gaussian Mixture Models, In: S.Z. Li, A.K. Jain (Eds.), Encyclopedia Of Biometrics, Springer, Boston, MA, 2015, Https://Doi.Org/10.1007/978-1- 4899- 7488-4_196.
[14] Stauffer And W. E. L. Grimson., "Adaptive Background Mixture Models For Real Time Tracking", International Conference On Computer Vision And Pattern Recognition, 2,1999.
[15] Butler, Darren E And Bove, V Michael And Sridharan, Sridha. Real-Time Adaptive Foreground/Background Segmentation. EURASIP Journal On Advances In Signal Processing 2005.
[16] Butler, Darren E And Bove, V Michael And Sridharan, Sridha. Real-Time Adaptive Foreground/Background Segmentation. EURASIP Journal On Advances In Signal Processing 2005.
[17] Ha SV-U, Chung NM, Phan HN, Nguyen CT. Tensormog: A Tensor-Driven Gaussian Mixture Model With Dynamic Scene Adaptation For Background Modelling. Sensors. 2020; 20(23):6973.
[18] Popović B, Cepova L, Cep R, Janev M, Krstanović L. Measure Of Similarity Between Gmms By Embedding Of The Parameter Space That Preserves KL Divergence. Mathematics. 2021; 9(9):957.
[19] Lee, Hj., Cho, S. (2004). Combining Gaussian Mixture Models. In: Yang, Z.R., Yin, H., Everson, R.M. (Eds) Intelligent Data Engineering And Automated Learning – IDEAL 2004. IDEAL 2004. Lecture Notes In Computer Science, Vol 3177. Springer, Berlin, Heidelberg. Https://Doi.Org/10.1007/978-3-540-28651- 6_98
[20] P. -M. Jodoin, L. Maddalena, A. Petrosino And Y. Wang, "Extensive Benchmark And Survey Of Modeling Methods For Scene Background Initialization," In IEEE Transactions On Image Processing, Vol. 26, No. 11, Pp. 5244-5256, Nov. 2017, Doi: 10.1109/TIP.2017.2728181.
[21] Atev S, Masoud O, Papanikolopoulos N. Practical Mixtures Of Gaussians With Brightness Monitoring. IEEE Conf On Intt Transportation Systems, Proceedings (ITS 2004),2004; 423-428.
[22] Zang Q, Klette R. Parameter Analysis For Mixture Of Gaussians. CITR Technical Report 188, Auckland University, 2006.
[23] Popović B, Cepova L, Cep R, Janev M, Krstanović L. Measure Of Similarity Between Gmms By Embedding Of The Parameter Space That Preserves KL Divergence. Mathematics. 2021; 9(9):957.
[24] Stauffer, C., & Grimson, W.E.L. (1999). Adaptive Background Mixture Models For Real-Time Tracking. Proceedings Of The IEEE Computer Society Conference On Computer Vision And Pattern Recognition (CVPR), 246-252.
[25] Chandola, V., Banerjee, A., & Kumar, V. (2009). Anomaly Detection: A Survey. ACM Computing Surveys, 41(3), 15:1–15:58.
[26] A. Tarafdar, S. Roy, A. Mondal, R. Sen And A. Adhikari, "Image Segmentation Using Background Subtraction On Colored Images," 2019 International Conference On Opto-Electronics And Applied Optics (Optronix), Kolkata, India, 2019, Pp. 1-4, Doi: 10.1109/OPTRONIX.2019.8862447.
[27] Zivkovic, Z. (2004). Improved Adaptive Gaussian Mixture Model For Background Subtraction. Proceedings Of The 17th International Conference On Pattern Recognition (ICPR), 2, 28-31.
[28] Young-Min Song, Seungjong Noh, Jongmin Yu, Cheon-Wi Park, And Byung-Geun Lee, Member,IEEE , Background Subtraction Based On Gaussian Mixture Model Using Color & Depth Information .
[29] Hajer Fradi, Jean-Luc Dugelay ," Robust Foreground Segmentation Using Improved Gaussian Mixture Model And Optical Flow" ,2014.
[30] Benjamin Langmann, Seyed E. Ghobadi, Klaus Hartmann, Otmar Loffeld , " MULTI-MODAL BACKGROUND SUBTRACTION USING GAUSSIAN MIXTURE MODELS", ZESS - Center For Sensor Systems, University Of Siegen , France, September 1-3, 2010.
[31] RENXI CHEN1 , XINHUI LI2 , And SHENGYANG LI3.(November 2020) , "A Lightweight CNN Model For Refining Moving Vehicle Detection FromSatellite Videos".