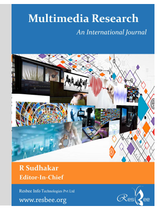
Vol.7 No.3 July 2024

Multimedia Research

Received 14 May, Revised 24 June, Accepted 26 July



LeNet Architecture and IEKF-KCF Scheme for Object Detection and Tracking

Kavya Jasti
University of Missouri Kansas, Missouri.
jastikavya99@gmail.com

Abstract: Object detection and tracking are vital components of computer vision, with applications ranging in fields such as autonomous vehicles, surveillance, robotics, and augmented reality. Object detection focuses on recognizing and pinpointing objects within images or video, while tracking is responsible for following their movement through sequential frames. Deep learning advancements have significantly improved the performance and accuracy of these tasks. However, obstacles like changing object appearances, and occlusion in real-time systems continue to pose challenges. This paper proposes a novel Object detection and tracking system to analyze visual data and provide accurate predictions. The proposed work adopts an Eigen Value Decomposition-based Canny Edge Detection (EVD-CED) approach that acquires the image edges; conversely, a Spherical object detection technique is employed that extracts the position of the image. Further, the image edges and image position are combined, then the fused image is generated to detect the object. Further, an innovative model called LeNet architecture recognizes the particular position of the moving target. Once the position of the object is detected, the object tracking process is carried out using the Improved Extended Kalman Filter-based Kernel Correlation Filter Tracking (IEKF-KCF) scheme. Thereby, the proposed work accurately detects the object and performs the tracking process efficiently. The IEKF-KCF attained the IOU of 0.854, NIOU of 0.827, MOTP of 0.844, and MAP of 0.814, respectively.

Keywords: Object detection, object tracking, LeNet, IEKF-KCF scheme and EVD-CED.

Nomenclature

Abbreviation	Description
EVD-CED	Eigen Value Decomposition-based Canny Edge Detection
HOTA	Higher Order Tracking Accuracy
IEKF-KCF	Improved Extended Kalman Filter-based Kernel Correlation Filter Tracking
IOU	Intersection Over Union
IDFP	Identify False Positive
IDFN	Identify False Negative
VOT	Visual Object Tracking
ROI	Region of Interest
CNN	Convolutional Neural Networks
AIA-IFRCNN	Automated Image Annotation with Inception v2-based Faster RCNN
CCTV	Closed-Circuit Television
UAV	Unmanned Aerial Vehicle
DCF-CSRT	Discriminative Correlation Filter-Channel and Spatial Reliability Tracker
MAP	Mean Average Precision
MOTA	Multi Object Tracking Accuracy
MOTP	Multi Object Tracking Precision
NIOU	Normalize Intersection Over Union
CSRT	Channel Spatial Reliability Tracker
KCF	Kernelized Correlation Filter Tracker

1 Introduction

Object recognition and tracking in autonomous aerial vehicles is a highly challenging task, as it requires accurate, agile detection and tracking in real-time, all while maintaining reasonable energy consumption. Various approaches have been developed for object tracking [4] [6], utilizing different types of sensors. Among them, vision-based object detection [1] [2], particularly using sensors like RGB cameras, is one of the most cost-effective and practical solutions. Additionally, the data captured by vision sensors can

https://doi.org/10.46253/j.mr.v7i3.a5

support other tasks, such as odometry and navigation. However, the major limitation of vision-based tracking methods lies in their high computational demands, which can impact performance, energy efficiency, and accuracy.

VOT involves recognizing a target, represented by an ROI, within a video. It has a wide range of applications, including multimedia, robotics, augmented reality, security and surveillance, and entertainment, all of which rely on effective tracking. Despite advances in the field, particularly with CNNs [3] [5], tracking remains a critical task in computer vision. Typically, tracking algorithms create a model of the target ROI and use this model to locate the target in subsequent frames. These algorithms must also handle challenges such as occlusions, rapid target movements, and situations where the target moves out of view. Another crucial factor is tracker speed, referring to the time taken to identify the target in every frame, as real-world applications demand efficient per-frame processing times.

Recently, CNNs have demonstrated greater effectiveness in various computer vision tasks [7], such as image classification, object detection, and semantic segmentation. This success is attributed to CNNs capability to extract semantically meaningful representations from visual details. As a result, CNNs have been increasingly applied to tracking tasks. Typically, CNN-based trackers filter convolutional features of targets from consecutive frames and use a cross-correlation technique between frames to locate the target. These methods are often demand large amounts of data and time-consuming. Although some DL techniques can deliver both real-time performance and accuracy, they still necessitate specialized hardware as well as they consume substantial power to achieve high speeds. Therefore, traditional tracking methods [9] may not consistently guarantee high detection rates in all scenarios. Hence, this paper proposes a novel Object detection and tracking system [8] [10] to analyze visual data and provide accurate predictions. The contribution of this work is as follows: Contributing EVD-CED approach that eliminates noise and details to extract the image edge and proposing IEKF-KCF technique for object tracking.

The remaining section is organized as follows: Section 2 reveals the literature review on object detection and tracking. Section 3 defines the proposed methodology. Section 4 analyzes the results over other approaches and Section 5 summarizes the conclusion.

2 Literature Review

In 2023, Z. Meng, *et al.* [1] implemented a HYDRO-3D approach that improved object detection by integrating historical tracking information. It utilized features from the V2X-ViT detection algorithm, combining them with tracking data to enhance inference accuracy. Moreover, a new spatial-temporal 3D NN processed local and global aspects of this historical data, creating a feature map that strengthens object detection performance.

In 2023, T. Keawboontan and M. Thammawichai, [2] devised a real-time DL model for multiple object tracking in UAV aerial videos. By combining tracking and detection through adjacent frame pairs, the proposed approach addressed class imbalance with a multi-loss function and reduced computational time. Furthermore, the suggested dual regression bounding box method improved frame association, enabling object ID verification and online tracking through accurate future position predictions.

In 2021, Q. Yu, *et al.* [3] executed an object detection-tracking algorithm using a radar-photoelectric system. The strategy started with a first-frame object extraction method to identify the tracking object, followed by an ROI prediction algorithm to detect objects within predicted ROIs. This approach effectively addressed challenges in marine tests and ensured stable extraction of radar-guided objects, even with multiple similar objects in view.

In 2024, Vijiyakumar, K., *et al.* [4] implemented a novel model AIA-IFRCNN. This model used the DCF-CSRT method for image explanation, Inception v2 for feature extraction, and Faster RCNN for object detection, and then performed softmax layer for classification. Furthermore, the outcomes revealed that AIA-IFRCNN outperformed traditional schemes in tracking and detection accuracy.

In 2024, Lago, Allan, *et al.* [5] devised a low-cost, lightweight, modular approach for real-time object detection and tracking. By combining real-time detection schemes with reasonable embedded hardware, the proposed system geolocated detected objects using image metadata that minimized computational overhead. Additionally, the algorithm enhanced accuracy by filtering geolocated detections with a clustering strategy to eliminate false positives.

3 An Overview of Object Detection

In computer graphics, object detection is a key task that involves predicting the location and category of an object. Challenges arise when determining an object's position in images captured by CCTV cameras. Object detection determines the procedure of identifying and locating objects within images or video frames, determining their classes, positions, and boundaries. Tracking, on the other hand, involves monitoring the measure of detected objects over time across a sequence of frames. This necessitates robust strategies, which can handle difficulties including varying lighting conditions, occlusions, and changes in object appearance. Hence, this paper proposes a novel Object detection and tracking system to analyze visual data and provide accurate predictions, as shown in Fig. 1.

Initially, it adopts an EVD-CED approach that extracts the image edges; conversely, the Spherical object detection technique is employed that extracts the position of the image. Further, the image edges and image position are fused to detect the object. A pioneering CNN model called LeNet architecture extracts the specific position of the moving target. Once the position of the object is detected, the object tracking process ahead using the IEKF-KCF scheme. Thereby, the proposed work accurately detects the object and performs the tracking process efficiently.

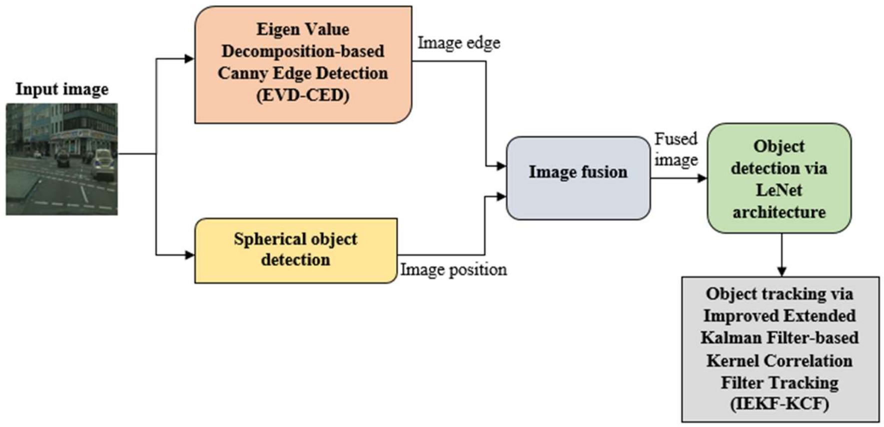


Fig. 1. *Architecture of proposed work*

3.1 Eigen Value Decomposition-based Canny Edge Detection (EVD-CED)

Canny edge detection [11] is a multi-step algorithm that has progressively refined its evaluation criteria in theory. The main steps of this algorithm include the following steps: "Gaussian filtering, calculating gradient magnitude and direction, non-maximum suppression, and double thresholding". Gaussian filtering effectively reduces noise, while gradient direction and magnitude help retrieve edge information from the image. Non-maximum suppression retains fine details of the edges by discarding non-significant values, and double thresholding distinguishes true edges from noise. However, it does not capture edges of distinct shapes and sizes at different scales. The non-maximum suppression leads to the loss of finer details. To avoid this, the EVD-CED method is proposed that eliminate noise and details and enhance the detection accuracy of the edge. The proposed EVD-CED method performs the following steps:

Step 1-Gaussian filtering: through this filtering technique, the input image can be smoothened, and remove noise from the image.

Step 2-Compute gradient magnitude: Along the X and Y directions, the magnitude and direction of the gradient are acquired from the Gaussian filter.

Step 3-Apply EVD method: Perform the EVD method to the gradient magnitude matrix, which provides Eigen value and Eigen vector that signifies the features of the image.

Let's consider, the gradient magnitude matrix as $A = \begin{bmatrix} 0 & 1 \\ -2 & -3 \end{bmatrix}$. To compute the eigenvalue and Eigenvector, need to solve the following formulation as in Eq. (1), where, V refers to Eigenvector and λ refers to Eigenvalue.

$$A * V = \lambda * V \tag{1}$$

Let's start by computing eigenvalues,

To find eigenvalue, solve the following derivations as in Eq. (2-6). Here, I indicates the identity matrix.

$$\left|(A - \lambda I)\right| = 0 \tag{2}$$

$$\left| \begin{bmatrix} 0 & 1 \\ -2 & -3 \end{bmatrix} - \lambda \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \right| = 0 \tag{3}$$

$$\left| \begin{bmatrix} -\lambda & 1 \\ -2 & -3-\lambda \end{bmatrix} \right| = 0 \tag{4}$$

$$(-\lambda)(-3-\lambda) + 2 = 0 \tag{5}$$

$$\lambda^2 + 3\lambda + 2 = 0 \tag{5}$$

$$(\lambda + 1)(\lambda + 2) = 0 \tag{6}$$

We get, $\lambda_1 = -1,\ \lambda_2 = -2$.

Now let's compute the corresponding Eigenvector for each eigenvalues. For $\lambda_1 = -1$, to compute the Eigenvector using the following formulation as in Eq. (7-11).

$$(A - \lambda_1 I)V_1 = 0 \tag{7}$$

$$\begin{bmatrix} 0 & 1 \\ -2 & -3 \end{bmatrix} - (-1)\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}\begin{bmatrix} X_1 \\ X_2 \end{bmatrix} = 0 \tag{8}$$

$$\begin{bmatrix} 0 & 1 \\ -2 & -3 \end{bmatrix} + \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}\begin{bmatrix} X_1 \\ X_2 \end{bmatrix} = 0 \tag{9}$$

$$\begin{bmatrix} 1 & 1 \\ -2 & -2 \end{bmatrix}\begin{bmatrix} X_1 \\ X_2 \end{bmatrix} = 0 \tag{10}$$

$$X_1 + X_2 = 0\ ;\ -2X_1 - 2X_2 = 0 \tag{11}$$

We obtain, $X_1 = -X_2 \quad ; X_1 = -X_2$.

Therefore, the Eigenvector corresponding to the eigenvalue $\lambda_1 = -1$ is expressed as in Eq. (12).

$$V_1 = \begin{bmatrix} +X_1 \\ -X_1 \end{bmatrix} \tag{12}$$

Assume $X_1 = 1$, then $V_1 = \begin{bmatrix} +1 \\ -1 \end{bmatrix}$.

For $\lambda_2 = -2$, to compute the Eigenvector using the following formulation as in Eq. (13-17).

$$(A - \lambda_2 I)V_2 = 0 \tag{13}$$

$$\begin{bmatrix} 0 & 1 \\ -2 & -3 \end{bmatrix} - (-2)\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}\begin{bmatrix} X_1 \\ X_2 \end{bmatrix} = 0 \tag{14}$$

$$\begin{bmatrix} 0 & 1 \\ -2 & -3 \end{bmatrix} + \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix}\begin{bmatrix} X_1 \\ X_2 \end{bmatrix} = 0 \tag{15}$$

$$\begin{bmatrix} 2 & 1 \\ -2 & -1 \end{bmatrix}\begin{bmatrix} X_1 \\ X_2 \end{bmatrix} = 0 \tag{16}$$

$$2X_1 + X_2 = 0\ ;\ -2X_1 - X_2 = 0 \tag{17}$$

We get, $2X_1 = -X_2 \quad ; 2X_1 = -X_2$.

Therefore, the Eigenvector corresponding to the eigenvalue $\lambda_2 = -2$ is expressed as in Eq. (18).

$$V_2 = \begin{bmatrix} +X_1 \\ -2X_1 \end{bmatrix} \tag{18}$$

Let's assume $X_1 = 1$, then $V_2 = \begin{bmatrix} +1 \\ -2 \end{bmatrix}$.

Finally, the eigenvalue decomposition of matrix A is given in Eq. (19). Here, D refers to the diagonal matrix.

$$A = pDp^{-1} \tag{19}$$

$$A = \begin{bmatrix} 1 & 1 \\ -1 & -2 \end{bmatrix}\begin{bmatrix} -1 & 0 \\ 0 & -2 \end{bmatrix}\begin{bmatrix} 1 & 1 \\ -1 & -2 \end{bmatrix}^{-1} \tag{20}$$

$$A = \begin{bmatrix} 1 & 1 \\ -1 & -2 \end{bmatrix}\begin{bmatrix} -1 & 0 \\ 0 & -2 \end{bmatrix}\begin{bmatrix} 2 & 1 \\ -1 & -1 \end{bmatrix} \tag{21}$$

Step 4-Double threshold processing: Recognizing and relating edges with a Double thresholding strategy.

Thereby, this proposed approach detects several kinds of edges like weak and strong edges and it can precisely locate the position of the edges. The acquired edge images are represented as I^{Edge}.

3.2 Spherical Object Detection

Spherical Object Detection [12] is the process of identifying and localizing spherical-shaped objects in images or videos through the use of computer vision and image processing techniques. This approach enables the precise identification and localization of spherical objects across various domains, effectively providing their positions within the image. The acquired edge images are represented as I^{Sph}.

3.3 Image Fusion

In this step, the edge image, I^{Edge} obtained from EVD-CED, and the position image, I^{Sph} obtained from Spherical object detection are fused [15] together. The pixels of the major and minor diagonals are organized into rows of the diagonal image, starting from the top row, with the pixels of the minor diagonals positioned in the center of their respective rows. It is important to note that the diagonal image matrix will have dimensions of $(m+n-1)\times Min(m,n)$. To obtain a properly fused image, we resize this diagonal image matrix to dimensions of $m\times n$. Finally, we combine the original image matrix with this newly created diagonal image matrix to produce a resultant fused image, I^{Fuse} matrix of size $m\times n$.

3.4 Object Detection using LeNet

The LeNet architecture [13] is a pioneering convolutional neural network (CNN) model that is adopted for object detection, which extracts the specific position of the moving target. This architecture comprises several layers, beginning with an input layer that accepts fused images, I^{Fuse}. The first layer is a convolutional layer that applies six filters (or kernels) to the fused image, I^{Fuse} resulting in a set of six feature maps. Each of these feature maps is then processed through a nonlinear activation function, typically the sigmoid or hyperbolic tangent function, to introduce non-linearity into the model.

Following the convolutional layer, a subsampling layer, often referred to as a pooling layer, is applied using average pooling. This layer down-samples the feature maps, reducing their dimensionality while preserving essential spatial information. The architecture includes a second convolutional layer that utilizes sixteen filters to extract additional features, followed by another pooling layer for further down-sampling. After the feature extraction layers, the output is flattened into fully connected layers, which integrate the high-level features learned from the prior layers. The architecture concludes with a softmax layer that generates probabilities for each class, allowing the model to detect the object.

3.5 Object Tracking via Improved Extended Kalman Filter-based Kernel Correlation Filter Tracking (IEKF-KCF)

After the object detection process, the object is tracked using the IEKF-KCF approach, which is the extension of the EKF [14]. This approach represents a recursive process that can linearize the nonlinear scheme over the current covariance and mean. The proposed IEKF-KCF algorithm follows the below steps:

Step 1- Initialization: Let's consider $Z_0 = Z_{initial}$ and $\hat{a}_0 = a_0$.

Step 2- Prediction step: This step updates the time equation. The prediction of the state vector $\hat{a}_k^-$ is defined as in Eq. (22), where, $\hat{a}_{k-1}^-$ indicates the estimated state vector at the prior time step; and b_{k-1} denotes the control vector at the prior time step.

$$\hat{a}_k^- = f\left(\hat{a}_{k-1}^-, b_{k-1}\right) \tag{22}$$

The main limitation of that it can assist the target process or track the target through without constantly updating the state of the target. To tackle this, an improved correlation is estimated between the state vector and control vector. Traditionally, Pearson correlation [16] is applied that can be computationally expensive in complex environments. Then, the improved correlation can be formulated as in Eq. (23).

$$_{New}\gamma = \left[\frac{\sum\left(c_i-\bar{c}\right)\left(1-\ln\left(1+e^{\bar{c}}-1\right)+\left(d_i-\bar{d}\right)\left(1-\ln\left(1+e^{\bar{d}}-1\right)\right)\right)}{\sqrt{\sum\left(c_i-\bar{c}\right)^2+\left[\frac{\pi}{n}\left\{10\sin^2\left(c_i\bar{c}\right)\right\}\right]}+\sqrt{\sum\left(d_i-\bar{d}\right)^2+\left[\frac{\pi}{n}\left\{10\cos^2\left(d_i\bar{d}\right)\right\}\right]}}\right] \tag{23}$$

Further, estimate the priori covariance matrix as in Eq. (24).

$$Z_k^- = Z_{k-1}\left(Z_{cd}\right)^{-1} \tag{24}$$

Step 3- Correction step: This step updates the measurement equation by computing the filter gain as in Eq. (25). The state computation of $\hat{a}_k$ is estimated as in Eq. (26) and the posteriori covariance is computed as in Eq. (27).

$$D_k = Z_k^- H_k^T \left(H_k^T Z_k^- H_k^T + B_k\right)^{-1} \tag{25}$$

$$\hat{a}_k = \hat{a}_k^- + D_k\left(m_k - h\left(\hat{a}_k^-, 0\right)\right) \tag{26}$$

$$Z_k = \left(I - D_k H_k\right) Z_k^{-} \tag{27}$$

Thereby, the object tracking process is conducted effectively using the IEKF-KCF approach.

4 Results and Discussion

4.1 Simulation Procedure

The proposed Object Detection and Tracking was implemented using Python, precisely “Version 3.7”. The processor employed was “Intel(R) Core(TM) i5-4210U CPU @ 1.70GHz 1.70 GHz and the Installed RAM size was 8.00 GB." Further, the object detection and tracking was evaluated using the Cityscapes Image Pairs dataset [17].

4.2 Dataset Description

This dataset encompasses of 2975 training and 500 validation image files. The image size presented in this file was 256x512 pixels. Moreover, every file was a combination of the original image on the left side of the image and the labeled image (outcome of semantic segmentation) on the right side.

4.3 Performance Analysis

The comparative evaluation was performed to gauge the IEKF-KCF approach, highlighting its assessment with established strategies for Object Detection and Tracking. This investigation employed a wide-ranging type of performance measures, such as HOTA, MOTA, IDFP, IOU, IDFN, MAP, MOTP, and NIOU to determine the IEKF-KCF method's efficacy. Moreover, an ablation study, computation Time, and statistical analysis were accomplished. The IEKF-KCF approach was contradicted by conventional methods, such as the KCF track, Median flow track, CSRT, and conventional Extended Kalman. In addition, the original images and Improved EKF Object tracking images are shown in Fig. 2.

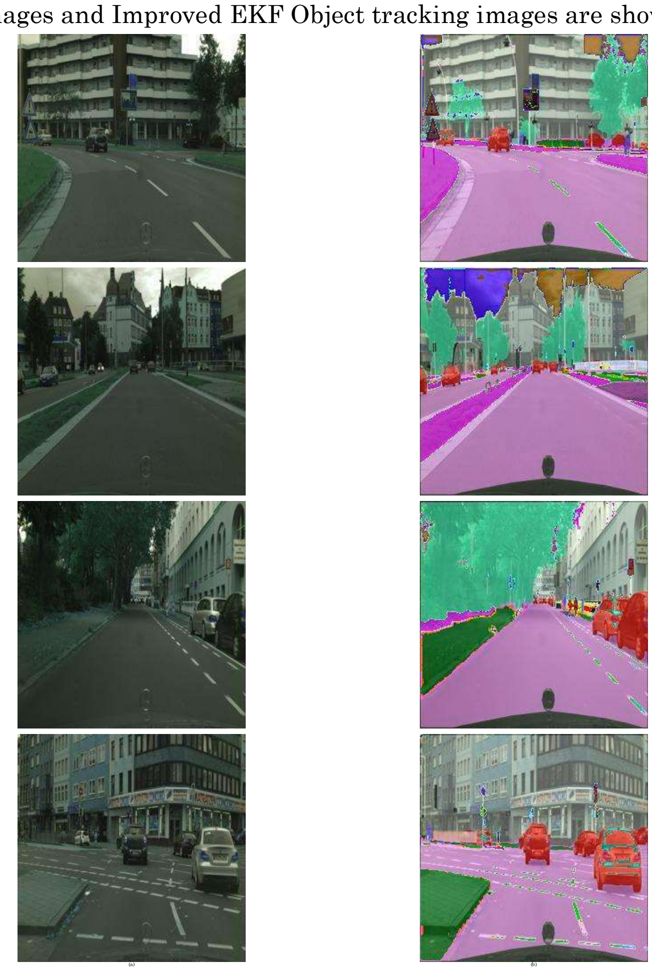


Fig. 2 *Images for Object Tracking a) Original Images and b) Improved Extended Kalam Filter based Object tracking images*

4.4 Comparative Analysis of HOTA, IDFN, IDFP, and IOU

Fig. 3 presents the assessment of the IEKF-KCF strategy in comparison with the KCF track, Median Flow track, CSRT, and Conventional Extended Kalman for Object Detection and Tracking. The evaluation is conducted for HOTA, IDFN, IDFP, and IOU metrics. Our main objective is to maximize these values for effective object detection and tracking. For 90% of training data, the IEKF-KCF strategy acquired the highest HOTA of 90.756, indicating its superior performance in object tracking. In contrast, the traditional methods like the KCF track, Median flow track, CSRT, and Conventional Extended Kalman obtained lesser HOTA values. The highest IDFN scored by the IEKF-KCF approach is 85.112 at 60% of training data. While the training data developed, the IDFN values gradually improved. With 70%, 80%, and 90% of training data, the IEKF-KCF strategy scored maximal IDFN of 86.167, 86.967, and 87.513, respectively. Similarly, the IEKF-KCF model generated maximal IDFP values compared to conventional approaches like the KCF track, Median Flow track, CSRT, and Conventional Extended Kalman. This upgrading is ascribed to EVD-CED that can be capable of eliminating noise and details, which enhances the detection accuracy of the edges. Accurate object detection is done through the LeNet framework. In addition, IEKF is employed to effectively develop the quality of target tracking.

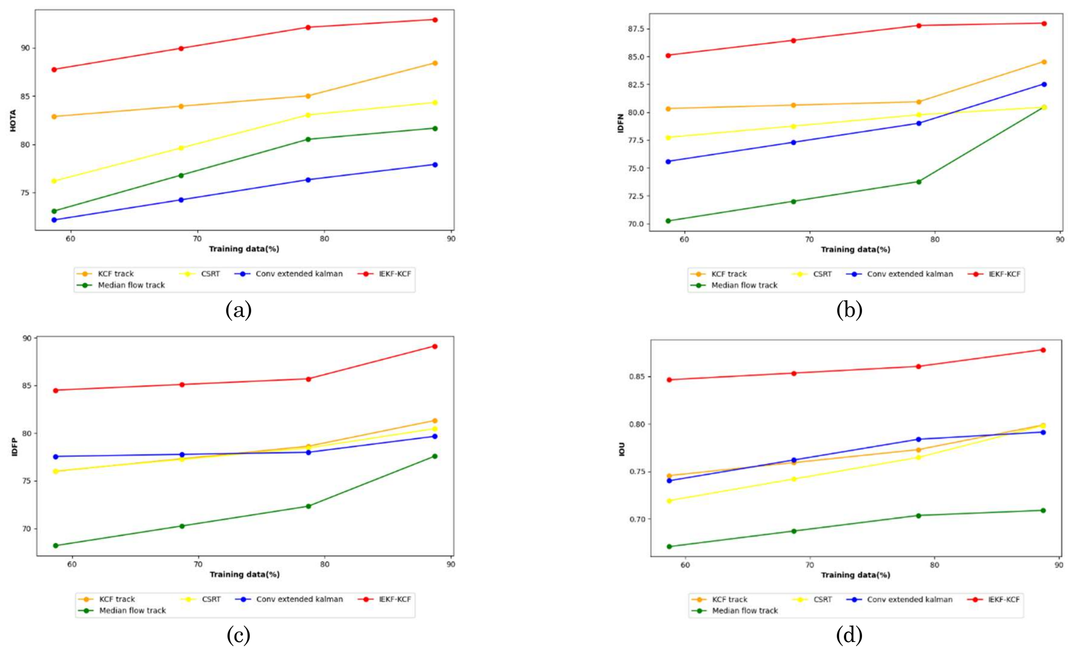


Fig. 3. *Assessment on IEKF-KCF and Conventional methods a) HOTA b) IDFN c) IDFP and d) IOU*

4.5 Comparative Analysis of MAP, MOTA, MOTP, and NIOU

The assessment of the IEKF-KCF model with conventional approaches, including the KCF track, median Flow track, CRT, and Conventional Extended Kalman for object detection and tracking is depicted in Fig. 4. Performance metrics like MAP, MOTA, MOTP, and NIOU are employed to analyze the efficacy of the models. The values of these metrics are greater for effective object tracking. While examining the MAP metric, the worst performance is observed in Conventional Extended Kalman with a value of 0.731 in the training data 90%, this value is closely aligned with the Median Flow track of 0.733. The KCF track and CSRT scored the MAP of 0.775 and 0.776, exhibiting slightly enhanced performance. However, the IEKF-KCF strategy recorded the greatest MAP of 0.825. Simultaneously, the IEKF-KCF approach achieved the highest MOTA of 91.246 (training data=90%), whereas the KCF track, Median Flow track, CSRT, and Conventional Extended Kalman attained the least MOTA values. The EVD-CED can detect distinct types of edges and it can exactly detect the position of the edges. In addition, the employment of IEKF lessens the object boundary tracking fault and narrows the possibility of candidate tracking region.

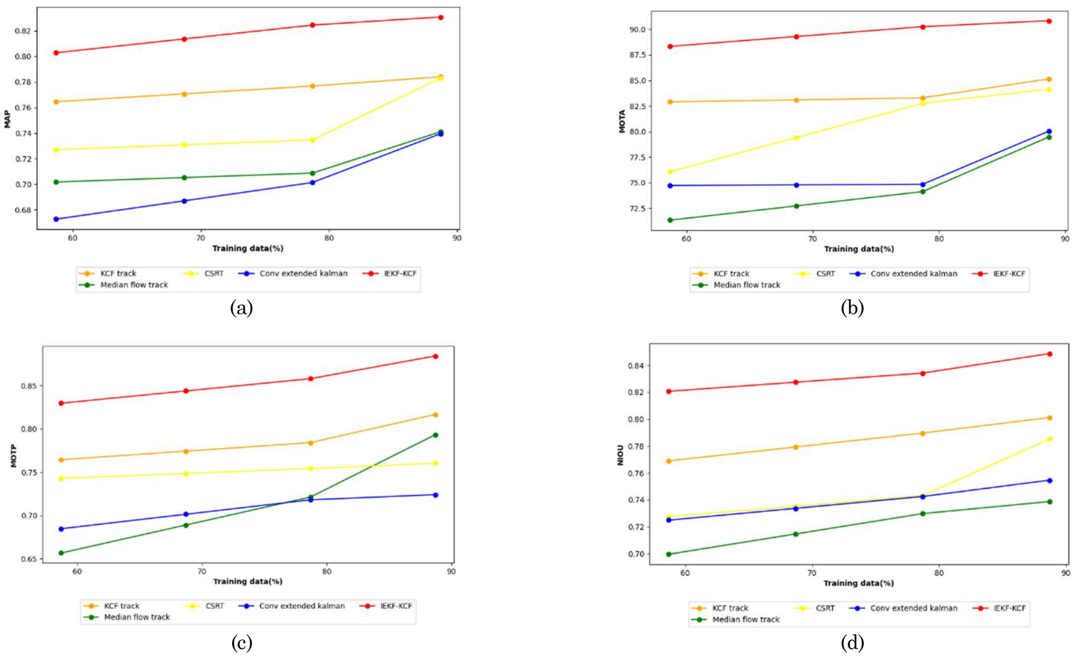


Fig. 4. *Assessment on IEKF-KCF and Conventional methods a) MAP b) MOTA c) MOTP and d) NIOU*

4.6 Statistical Analysis of HOTA

In the realm of object detection and tracking, a statistical assessment conducted on IEKF-KCF strategy contradicted with KCF track, Median flow track, CSRT, and Conventional Extended Kalman is summarized in Table I. In terms of Maximum Statistical metric, the IEKF-KCF approach scored the highest HOTA of 92.918, while the conventional approaches attained least HOTA, such as the KCF track at 88.417, Median Flow Track at 81.670, CSRT at 84.330 and Conventional Extended Kalman at 77.918, respectively. The introduction of EVD-CED and Improved Extended Kalam Filter makes the IEKF-KCF approach more efficacious in object tracking.

Table 1. *Statistical Evaluation on HOTA*

Methods	Standard Deviation	Minimum	Mean	Maximum	Median
KCF track	2.078	82.879	85.063	88.417	84.479
Median flow track	3.363	73.098	78.023	81.670	78.661
CSRT	3.163	76.199	80.796	84.330	81.328
Conventional Extended Kalman	2.164	72.173	75.171	77.918	75.296
IEKF-KCF	2.010	87.755	90.678	92.918	91.020

4.7 Ablation Study on IEKF-KCF

In the ablation study, it is worthwhile to recognize the impact of each innovation individually. To measure the impact of specific components, researchers recurrently investigate their models with each of them disabled and quantify the decline of overall model performance. The ablation analysis of the IEKF-KCF scheme in comparison to two distinct variants, like IEKF-KCF without Canny Edge Detection and IEKF-KCF with Conventional Canny Edge Detection is exposed in Table II. The NIOU of the IEKF-KCF scheme is 0.827, while the IEKF-KCF without canny edge detection and IEKF-KCF with conventional canny edge detection scored the least NIOU of 0.795.

Table 2. *Ablation Evaluation on IEKF-KCF Approach, IEKF-KCF without Canny Edge Detection, and IEKF-KCF with Conventional Canny Edge Detection*

Metrics	IEKF-KCF without canny edge detection	IEKF-KCF with conventional canny edge detection	IEKF-KCF
IOU	0.776	0.796	0.854
NIOU	0.751	0.795	0.827
MOTP	0.750	0.797	0.844
MAP	0.736	0.784	0.814
MOTA	82.076	85.613	89.292
IDFN	78.087	82.731	86.456
IDFP	78.034	80.253	85.101
HOTA	81.727	85.317	89.932

4.8 Analysis on Computation Time

The Computation Time analysis on the IEKF-KCF strategy is compared with traditional methodologies for object detection and tracking and is presented in Table III. The least computation time attained using the IEKF-KCF approach is 30.301s, while the KCF track, Median flow track, CSRT, and Conventional Extended Kalman generated lesser computation time, ranging from 34.993s to 43.444s.

Table 3. *Computation Time Analysis on IEKF-KCF and Conventional Methods*

Methods	Computation time(s)
KCF track	35.449
Median flow track	43.444
CSRT	36.479
Conventional Extended Kalman	34.993
IEKF-KCF	30.301

5 Conclusion

This paper proposed a novel Object detection and tracking system to analyze visual data and provide accurate predictions. The proposed work adopted an Eigen Value Decomposition-based Canny Edge Detection (EVD-CED) approach that acquired the image edges; conversely, a Spherical object detection technique was employed that extracts the position of the image. Further, the image edges and image position were combined, and then the fused image was generated to detect the object. Further, an innovative model called LeNet architecture recognized the specific position of the moving target. Once the position of the object was detected, the object tracking process was carried out using the Improved Extended Kalman Filter-based Kernel Correlation Filter Tracking (IEKF-KCF) scheme. At 90% of training data, the proposed strategy acquired the highest HOTA of 90.756, indicating its superior performance in object tracking. In contrast, the traditional methods like KCF track, Median flow track, CSRT, and Conventional Extended Kalman obtained lesser HOTA values.

Compliance With Ethical Standards

Conflicts of interest: Authors declared that they have no conflict of interest.

Human participants: The conducted research follows ethical standards and the authors ensured that they have not conducted any studies with human participants or animals.

References

[1] Z. Meng, X. Xia, R. Xu, W. Liu, and J. Ma, "HYDRO-3D: Hybrid Object Detection and Tracking for Cooperative Perception Using 3D LiDAR," in IEEE Transactions on Intelligent Vehicles, vol. 8, no. 8, pp. 4069-4080, 2023.

[2] T. Keawboontan, and M. Thammawichai, "Toward Real-Time UAV Multi-Target Tracking Using Joint Detection and Tracking", in IEEE Access, vol. 11, pp. 65238-65254, 2023.

[3] Q. Yu, B. Wang, and Y. Su, "Object Detection-Tracking Algorithm for Unmanned Surface Vehicles Based on a Radar-Photoelectric System", in IEEE Access, vol. 9, pp. 57529-57541, 2021.

[4] Vijiyakumar, K., V. Govindasamy, and V. Akila. "An effective object detection and tracking using automated image annotation with inception based faster R-CNN model", International Journal of Cognitive Computing in Engineering, vol.5, pp.343-356, 2024.

[5] Lago, Allan, Sahaj Patel, and Aditya Singh. "Low-Cost Real-Time Aerial Object Detection and GPS Location Tracking Pipeline." ISPRS Open Journal of Photogrammetry and Remote Sensing, vol.13, 2024.

[6] J. Alikhanov, and H. Kim, "Online Action Detection in Surveillance Scenarios: A Comprehensive Review and Comparative Study of State-of-the-Art Multi-Object Tracking Methods," in IEEE Access, vol. 11, pp. 68079-68092, 2023.

[7] M. Rabah, A. Rohan, M. -H. Haghbayan, J. Plosila, and S. -H. Kim, "Heterogeneous Parallelization for Object Detection and Tracking in UAVs," in IEEE Access, vol. 8, pp. 42784-42793, 2020.

[8] M. Chen, S. Banitaan, and M. Maleki, "Enhancing Pedestrian Group Detection and Tracking Through Zone-Based Clustering," in IEEE Access, vol. 11, pp. 132162-132179, 2023.

[9] J. Chen, Z. Xi, C. Wei, J. Lu, Y. Niu, and Z. Li, "Multiple Object Tracking Using Edge Multi-Channel Gradient Model With ORB Feature," in IEEE Access, vol. 9, pp. 2294-2309, 2021.

[10] Abba, Sani, Ali Mohammed Bizi, Jeong-A. Lee, Souley Bakouri, and Maria Liz Crespo. "Real-time object detection, tracking, and monitoring framework for security surveillance systems." Heliyon, vol.10, no. 15, 2024.

[11] Xu, Jie, Sijie Niu, and Zhifeng Wang. "Object tracking method based on edge detection and morphology", EURASIP Journal on Advances in Signal Processing, vol.45, no. 1, 2024.

[12] Hansen, Jakob Grimm, and Rui Pimentel de Figueiredo, "Active Object Detection and Tracking Using Gimbal Mechanisms for Autonomous Drone Applications", Drones, vol.8, no. 2, 2024.
[13] Mahmoud, Shaimaa, Mohamed Gaber, Gamal Farouk, and Arabi Keshk, "Heart disease prediction using modified version of LeNet-5 model", Int. J. Intell. Syst. Appl, vol.14, no. 6, pp.1-12, 2022.
[14] Coelho, M. F., K. Bousson, and K. Ahmed, "An Improved Extended Kalman Filter for Radar Tracking of Satellite Trajectories", Designs, vol.5, no.54, 2021.
[15] Dey, Aniruddha, and Jamuna Kanta Sing, "An image fusion technique for efficient face recognition", In 2015 IEEE 2nd International Conference on Recent Trends in Information Systems (ReTIS), pp. 261-266, 2015.
[16] Roy, Anuradha, "Estimating correlation coefficient between two variables with repeated observations using mixed effects model", Biometrical journal, vol.48, no. 2, pp.286-301, 2006.
[17] https://www.kaggle.com/datasets/dansbecker/cityscapes-image-pairs