Bulletin of Electrical Engineering and Informatics
Vol. 15, No. 6, December 2026, pp. 5012∼5032
ISSN: 2302-9285, DOI: 10.11591/eei.v15i6.12903

Hybrid features for RGB-D object tracking using deep learning

Sari Awwad[1], Yousef Elsheikh[2], Ahmad Al-Rababa'a[3]
[1]Department of Computer Science and Applications, Faculty of Prince Al-Hussein bin Abdullah II of Information Technology, The Hashemite University, Zarqa, Jordan
[2]Department of Software Engineering, Faculty of Information Technology, Applied Science Private University, Amman, Jordan
[3]Department of Computer Science, Faculty of Information Technology, World Islamic Sciences and Education University (WISE), Amman, Jordan

Article Info

Article history:

Received Mar 26, 2026
Revised Sep 10, 2026
Accepted Oct 4, 2026

Keywords:

Convolutional neural networks
Deep learning
Feature fusion
Hybrid features
Kalman filter
Object tracking
RGB-D tracking

ABSTRACT

Tracking objects in red, green, blue, and depth (RGB-D) video remains challenging because visual appearance can change under occlusion, illumination variation, and motion blur, while depth frames may contain missing or noisy values. This study proposes a hybrid RGB-D object-tracking framework that combines convolutional neural network (CNN) features with local binary patterns (LBP) for texture and histogram of oriented gradients (HOG) for shape, using a Kalman filter and a deep appearance model to estimate object locations across frames. The method is evaluated on the Digital Twin Tracking Dataset (DTTD), comprising 100 videos with annotated ground-truth tracks. The experimental protocol uses an 80:20 training/testing split, 100 training epochs, a batch size of 32, and rotation and horizontal-flip augmentation. On the reported benchmark, the proposed approach achieves a success rate (SR) of 88.9%, a precision rate (PR) of 91.5%, and a root mean square error (RMSE) of 3.9 pixels, at 22 frames per second (FPS). These results indicate that combining learned semantic representations with handcrafted texture and shape descriptors improves tracking performance over the individual feature-based baselines, particularly under occlusion and illumination changes.

This is an open access article under the CC BY-SA license.

Corresponding Author:

Sari Awwad
Department of Computer Science and Applications
Faculty of Prince Al-Hussein bin Abdullah II of Information Technology, The Hashemite University
Zarqa 13133, Jordan
Email: sari@hu.edu.jo

1. INTRODUCTION

There is an increasing number of methods that utilize red, green, blue, and depth (RGB-D) information for object localization and tracking. RGB-D tracking systems have been studied for many years, and their applications are spreading to various fields including robotics, surveillance, autonomous systems, augmented reality, and human–computer interaction. They provide additional spatial information from RGB information, which enables accurate object separation and robust tracking even in the presence of visual distractions, background noise, and changes in lighting [1], [2]. Although RGB-D tracking still is a challenging task (e.g., due to sensor noise, missing depth values, changing backgrounds, occlusion, and fast motion), approaches are needed that are able to combine both modalities successfully and to exploit their synergies. In this sense, current RGB-D

tracking approaches can be roughly categorized into two groups, i.e., into deep learning-based methods that learn feature representation automatically and into methods for feature extraction using hand-crafted descriptors. Convolutional neural networks (CNNs) are able to efficiently train hierarchical feature representations of the input in order to achieve state-of-the-art results in many computer vision tasks [3], [4]. Even if sufficient information for object recognition can be extracted by deep learning features from images, the texture and shape information might be too fine-grained. Hence, besides the information that can be extracted by deep learning features, also information that is provided by hand-crafted features, like local binary patterns (LBP) or histograms of oriented gradients (HOG), can be very useful. This information is domain-specific and complementary to the high-level information, which is provided by CNNs [5], [6]. Recent work which integrates deep features with handcrafted features, achieves superior tracking performance, and enhanced environmental adaptability [1], [7]. Though recent years have seen great progress in the field of RGB-D tracking by using deep learning, so far, no sufficient discussion has been held on the use of deep features and hand-crafted features for a higher robustness in complex situations. The existing hybrid approaches use single features or only concatenate them to each other. Thus, they do not take into consideration the different characteristics of a feature. A combination of hybrid features and a motion prediction (e.g., by a Kalman filter) has not been investigated sufficiently yet.

Unlike previous hybrid approaches that use simple feature concatenation [1], [7] or rely solely on Fisher vector encoding [8], our ramework introduces the following four contributions: i) a multi-level fusion strategy that preserves modality-specific characteristics; ii) an adaptive weighting mechanism conditioned on depth map quality; iii) a motion-guided feature extraction approach using Kalman filter predictions; and iv) a systematic evaluation was conducted using the recently released Digital Twin Tracking Dataset (DTTD), which achieved the state-of-the-art tracking performance with an overall SR of 88.9% and an average PR of 91.5%. This integrated framework ensures that complementary features are combined in a principled manner, addressing the limitations of prior work where feature integration was performed without considering the different characteristics and quality of each feature type.

This paper is organized in the following way: in the following section 2 we will review the work of other researchers, using objects of interest for tracking in RGB-D images, combining different features, and the use of deep learning. In section 3 we will explain the data that we have used for our work, how we prepared it and what methods we have developed. Section 4 deals with our verification of the findings of section 3. Section 5 will contain a summary of this work, and also an overview of what we have added to the previous research, and where future research could be headed.

2. LITERATURE REVIEW

RGB-D based object tracking has recently received increasing attention and made great progress. In RGB-D based object tracking, the depth data generated by RGB-D cameras can effectively deal with the problem of lighting variation, background interference and object occlusion, and thus improve the object tracking performance. This survey presents a detailed survey of the recent RGB-D based object tracking methods, focusing on the methods that employ hybrid feature fusion, deep learning and feature selection.

2.1. Hybrid feature fusion in red, green, blue, and depth object tracking

Hybrid feature fusion that combines CNN deep features with hand-crafted features for RGB-D object tracking is effective. Amalgamating CNN features with hand-crafted features in a tracking system improves the tracking's dependability and robustness. The hybrid features supply additional insights which are helpful for improving the tracking's performance. This is demonstrated by the work of Wang *et al.* [1] in which a tracking system that consisted of a combination of CNN features and hand-crafted features was used. Loghmani *et al.* [7] proposed a recurrent convolutional fusion strategy for RGB-D based object recognition in. A recurrent convolutional fusion strategy was proposed for RGB-D based object recognition in [7]. The proposed model consists of two independent CNN paths for the RGB and depth images respectively. The feature fusion is implemented by using recurrent links in the proposed model to learn the temporal relation and to keep the features of all frames consistent. Li *et al.* [8] used a hybrid approach which combines CNN features with Fisher vector encoding to perform RGB-D object recognition and got the best recognition accuracy for most of the tested item categories. They achieved this by combining the best of both worlds i.e., the discriminative power of deep features and the compact representation power of Fisher vectors. By using a hybrid approach of both spatial and depth information, Chen and Zhou [2] achieved the highest detection accuracy for salient object detection in RGB-D images using their proposed hybrid-attention network which uses depth-aware features and spatial attention.

By using a feature selection mechanism, CNN-based features can be fused with handmade features to form multi-dimensional features for salient target detection, thereby achieving higher tracking performance than using individual features, according to Chen *et al.* [5]. They proposed a learning platform using CNN for choosing and fusing multi-dimensional features for salient target detection and demonstrated the merits of using feature selection mechanisms for tracking.

Although several feature representation methods can be merged to form a framework to achieve better object tracking, the specific processes for feature integration and the trade-off between deep and handmade features for the best feature integration still remain to be resolved in future research.

2.2. Deep learning approaches for red, green, blue, and depth object tracking

Recent work has employed deep learning to enhance tracking systems based on RGB-D data. Specifically, Zia *et al.* [3] compared state-of-the-art RGB-D object recognition using deep CNNs to traditional local descriptors. Their extensive evaluation demonstrated that deep features, such as those learned by CNNs, encode hierarchical representations which enable generalization across object categories as well as between vastly different environmental conditions. Sanchez-Riera *et al.* [4] employed deep learning to track hand movements from RGB-D sequences. By encoding hand pose priors via prior learning methods, the authors enabled robust tracking of the hand even in the presence of self-occlusion and fast motion. We demonstrated the consistency over frames of learned priors and corresponding tracking frameworks. Mou *et al.* [9] presented a new salient object detection method for RGB-D video sequences by making use of space-time features for optimal sequence tracking. By making use of the spatial and temporal information provided by the RGB-D video sequences, object detection was significantly improved for dynamic scenes. In the research community, for tracking RGB-D data, both recurrent as well as hierarchical methods have been proposed.

Li *et al.* [10] presented a hierarchical alternate interaction network for refining deep features in multiple layers for the task of RGB-D based salient object detection. The network alternates between refinement of deep features for RGB as well as depth streams in a hierarchical manner. Caglayan and Can [11] proposed a framework for object detection that utilized different feature representation layers. The use of hierarchical feature extraction improved the recognition accuracy in their proposed framework for salient object detection. Research using deep learning, especially hierarchical architectures, can successfully deal with the spatial-temporal information of RGB-D sensors to perform human tracking. These methods learn representations that are directly optimized for the tracking task.

2.3. Transformer-based red, green, blue, and depth tracking

Recent approaches based on transformers for visual tracking exploit self-attention mechanisms to model long-range dependencies within the input data. However, despite good performance reported by recent approaches such as OSTrack and TransT, current methods using transformers for tracking are hampered by several limitations. First, transformers' computational cost scales quadratically with the number of tokens (i.e., inputs), making them poorly suited for real-time applications. Second, the need for large training datasets and long training times is a barrier to adoption. Third, current transformers are not robust to domain shifts, meaning that their performance drops significantly when tracking is applied to sequences drawn from different distributions than those of the training data [12].

Our work on a CNN-LBP-HOG-Kalman framework for tracking gives us several advantages: i) efficiency-22 frames per second (FPS) on a single graphics processing unit (GPU), compared to transformer tracking approaches running at 8-15 FPS on similar hardware; ii) complementary features-semantic information from CNNs, plus texture/shape information from handcrafted features; iii) motion prior-the Kalman filter is modeling motion explicitly; and iv) practical deployment-our simple framework can be easily deployed on embedded systems.

2.4. Self-supervised learning for tracking

Learning visual representations without a lot of annotated data has been a challenge [13]. This work is also related to self-supervised learning, and in particular to learning visual representations with not a lot of annotated data. can improve tracking. Existing visual representation learning methods, however, suffer from many drawbacks, i) depth-specific challenges are not well explored. These are still not fully solved; ii) effective representation learning requires a large amount of unlabeled data; and iii) learned representations do not contain domain-specific information needed for tracking specific object types. may not have domain-specific features for certain object types.

However, there are several disadvantages of the existing visual representation learning methods; i) a number of depth related problems are still not resolved less explored; ii) require lots of unlabeled data to learn well and iii) representations the learned feature representations lack the domain-specific information needed to track specific types of objects. We propose a principled hybrid feature frame to complement self-supervised approaches. a self-supervised pre-training extension. One of the future work will be on self-supervised learning. In the future, the hybrid approach of features together with self-supervised learning may achieve better generalization.

2.5. Feature selection and fusion strategies

The integration method between selected features serves as a primary factor that boosts the capabilities of RGB-D object tracking. Tu *et al.* [14] developed a hybrid knowledge distillation network for RGB-D co-salient object detection that brings together RGB and depth information effectively. Their approach demonstrated that knowledge transfer between modalities improves overall detection performance. In the recent work, Ozcan and Cetin [15] proposed novel data fusion approach for human detection that utilizes information from RGB-D, thermal sensors, and other sensors in various situations. In their work, the novel data fusion approach is introduced by proposing a multi-modal fusion framework for improving human detection by using different sensor information. Also, similar ideas have been investigated and realized in various situations in several studies. For instance, Gao *et al.* [16] proposed a unified information fusion network, which can improve the object tracking, such as salient object, by using information from RGB-D images and RGB-T images, and showed robust results for various situations. Most research aims to improve tracking accuracy by combining deep learning techniques with handcrafted features that contain texture and shape descriptors. For example, research by Kamal and Jalal [6] a hybrid system was proposed for a depth sensor-based human detection and activity recognition. In this system both handcrafted and learned features were used for different activities. Research by Jiang *et al.* [17], object recognition and tracking for an indoor robot was done by using RGB-D sensor. The geometric and texture-based feature extraction improved the PR of the tracking.

2.6. Applications of red, green, blue, and depth tracking in various domains

RGB-D object tracking is also applied to human tracking, activity recognition, and to control systems that are to be made autonomous. For real-time human tracking using a single RGB-D camera, Xiao *et al.* [18] developed a deployment system for human tracking. To improve the accuracy of RGB-D video sequence people tracking, Xue *et al.* [19] combined deep learning with motion-related detection cues to robustly track people in crowded scenes. Khaire and Kumar [20] described human action recognition using deep learning on RGB-D data in their work and gave a survey of deep learning methods for human tracking and human interaction. Meanwhile, Schwarz *et al.* [21] introduced an integrated approach for object detection and for semantic segmentation of RGB-D data in cluttered environments for autonomous manipulation by robotics. Their method relies on accurate tracking to enable a robotic system to perform efficiently a task given to it. Jiang *et al.* [17] are able to improve object tracking for indoor robots by making use of RGB-D sensors to achieve the state-of-the-art result for object-position detection and for path-numeration. Recently a large designed RGB-D dataset for long-range object tracking was released in the form of the DTTD [22]. Thus tracking and perception functions of robotics vision systems were significantly improved.

2.7. Positioning of our approach

To clearly position our work in the current landscape, we summarize the key differentiators of our framework:

- Our framework is different from deep learning approaches [3], [4], [10] that use learned features. Explicitly uses handcrafted features to capture texture and shape information, which is lost in deep hierarchies of features.
- Our multi-level fusion with adaptive weighting is different from simple concatenation of features [1], [7]. Depth, quality and content of ethical modalities combination traits.
- Our method runs at 22 FPS, while the transformer based methods runs at 8-15 FPS [12], and easier deployment on resource constrained devices.
- Our approach requires less unlabeled data than self-supervised methods [13], implicitly exploits domain-specific features.
- We use Kalman filters to explicitly model motion and to augment appearance-based modeling.tracking, and better handling of motion discontinuities than learning-based approaches.

Our balanced solution that uses multiple paradigms to avoid their drawbacks. restricting themselves. This hybrid design is perfect for applications where computational resources, data availability, and real-time performance.

2.8. Future directions in red, green, blue, and depth object tracking

Recently, there have been tremendous advances in the deep learning and feature integration for object tracking. To enhance RGB-D based salient object detection using recursive dynamic filtering, a hierarchical approach was proposed by Pang *et al.* [23] for supporting object tracking. Recently, a survey of current state-of-the-art methods, existing challenges, and future possible research directions for RGB-depth based objects tracking were provided by Zhou *et al.* [24]. Very recently first results have been achieved for porting a transformer-based framework to RGB-D tracking in [12]. The self-attention network is the core component of this tracking framework, which enables the framework to learn long-range dependencies for robust tracking. Recently, powerful tools for self-supervised learning of images have been developed and we will experiment with them for image tracking, i.e., for tracking without the need for annotated data for learning to track and to improve generalization to new situations. In order to improve robustness and generalization of tracking models that are based on deep learning [25], recent research results also indicate that data augmentation plays a crucial role.

When training data is limited or extremely variable, large numbers of artificially generated training samples have to be processed by deep learning frameworks, which, themselves, have to be designed for large amounts of data [26], [27]. This issue is further complicated by the widespread usage of IoT in smart cities, requiring highly efficient and robust RGB-D tracking solutions, capable of operation in real environments, which are also deployable [28], [29]. For a long time, machine learning in general, and deep learning, in particular, have been successfully applied to various object tracking applications, including feature-based ones, by greatly improving tracking accuracy [30]. To tackle challenges of tracking in general, many recent approaches of hybrid prediction models have been designed to improve adaptability of learning in these models [31]. Very recently, the integration of RGB-D sensors into a single system for simultaneous object tracking and counting has been a subject of development of systematic frameworks for various applications, thereby indicating increasing maturity of tracking using RGB-D [32]. Our work includes improving the ways to fuse features, designing more efficient deep learning models, and making a comprehensive benchmark for RGB-D based object tracking. Future research on RGB-D tracking will be built on top of the deep learning methods and appropriate feature representations. In this way, the trackers are able to learn more features and more discriminative features from multimodal data. By combining deep learning with multimodal feature fusion, tracking accuracy, tracking robustness, and tracking adaptability can be significantly improved for various applications in real world.

3. METHOD

This section describes the data specifications, preprocessing operations, experimental setup for RGB-D object tracking based on hybrid features and deep learning.

3.1. Dataset

The DTTD [22] is used for the investigation. The dataset consists of a number of sequences, each with a number of RGB (color) and depth (D) frames for long-range tracking. The dataset is originally proposed in the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). The dataset: our dataset consists of 100 sequences, each with a number of RGB frames and a number of corresponding depth frames.

- 50 different objects. These include objects of daily use like books, cups and toys as well as more specific objects for particular applications like tools and electronics.
- Indoor/outdoor environments, various lighting, and differently complex backgrounds.
- Partial and full occlusion, as well as motion blur, objects of varying sizes and fast moving objects.
- RGB and depth frames are captured at regular time steps (to maintain a temporal stable view of objects) and in between frames to reduce the number of identical frames.
- Reliable ground truth annotations confirmed by measures of inter-annotator agreement.

The ground truth for the DTTD has been carefully annotated to include the exact position of objects in every frame. This allows for the use of the dataset in supervised learning, and allows for a very comprehensive evaluation of tracking algorithms.

3.2. Data preprocessing

Deep learning model is able to achieve better performance after going through the right data preprocessing steps to achieve consistency and best results in training process. Here are the steps involved in the preprocessing pipeline.

3.2.1. Image resizing

After the RGB and depth frames have been resized to 224×224 pixels (CNN input dimension), they can be fed into the CNN and–due to the batch-based processing–feature extraction can be done efficiently for all frames.

3.2.2. Normalization

Pixel intensity values are scaled to the 0-1 range to stabilize training gradients and to speed up convergence. The normalization is done as (1):

$$I_{\text{norm}} = \frac{I - I_{\min}}{I_{\max} - I_{\min}} \tag{1}$$

where I is the intensity value of the pixel in question and Imin, Imax are the minimum and maximum intensity values for the given image.

3.2.3. Data augmentation

We attempt to improve the model using different enhancement techniques in order to decrease overfitting.

- Rotation: ±15 deg. (It tries to rotate the object to handle different orientations of the object).
- Flip horizontal: 50%.

These methods allow for better generalization to unseen data. Deep learning-based tracking greatly benefits from data augmentation [25].

3.2.4. Handling missing and noisy depth data

The missing depth values of frames are filled by the interpolation using the neighbouring pixels, and the noisy depth frames are filtered by the median filtering to remove the sensor noise while keeping the edge information.

3.2.5. Depth quality analysis

To validate our depth preprocessing, we analyzed depth quality across the DTTD dataset as shown in Table 1. The comparisons have been done between different preprocessing settings (see Table 2). We selected nearest-neighbor interpolation for missing depth values because it preserves depth discontinuities better than bilinear or bicubic interpolation. A 5×5 median filter was empirically determined to be optimal.

Table 1. Depth quality analysis by sequence category

Sequence category	Missing depth%	Noisy pixels%	Avg depth std dev
Indoor well-lit	2.3±1.8%	1.5±0.9%	12.4±3.2
Indoor low-light	8.7±3.2%	5.2±1.8%	18.7±4.1
Outdoor daytime	4.1±2.1%	3.8±1.2%	15.3±3.8
Outdoor night	15.3±5.6%	8.9±3.1%	24.6±5.2
Overall	**7.6±5.1%**	**4.9±3.2%**	**17.8±5.6**

Table 2. Comparison of depth preprocessing settings

Preprocessing setting	SR (%)	PR (%)	Root mean square error (RMSE)	FPS
No preprocessing	73.2±1.5	76.8±1.6	7.2±0.4	24
Interpolation only	82.5±1.1	85.1±1.2	5.1±0.3	23
Median filter only	84.3±1.0	86.9±1.1	4.8±0.3	22
Interpolation+median	**88.9±0.7**	**91.5±0.8**	**3.9±0.2**	**22**
Gaussian filter	86.7±0.9	89.2±1.0	4.3±0.2	21
Bilateral filter	87.5±0.8	90.1±0.9	4.1±0.2	19

3.3. Proposed framework

In this work, the merits of the deep features learned by CNNs are combined with the robustness of the handcrafted features for tracking an object of interest in RGB-D video sequences. The proposed system, referred to as hybrid RGB-D object tracking system, is illustrated in the following framework.

a. Feature extraction: in the proposed approach, input RGB-D data are processed to extract complementary features, using a CNN and handcrafted feature descriptors, respectively.

b. Hybrid feature fusion: by combining the deep features with the handcrafted feature descriptors, the complementary information are integrated into a single feature space for the tracker.
c. Tracking mechanism: a Kalman filter, in combination with a deep appearance model, predicts the future motion of the object and ensures accurate tracking between subsequent frames.
d. Frame-by-frame object localization: in each frame of a video, object location is estimated using the merged features for updating the object location for tracking.

The hybrid feature extraction procedure for RGB-D object tracking is shown in Figure 1.

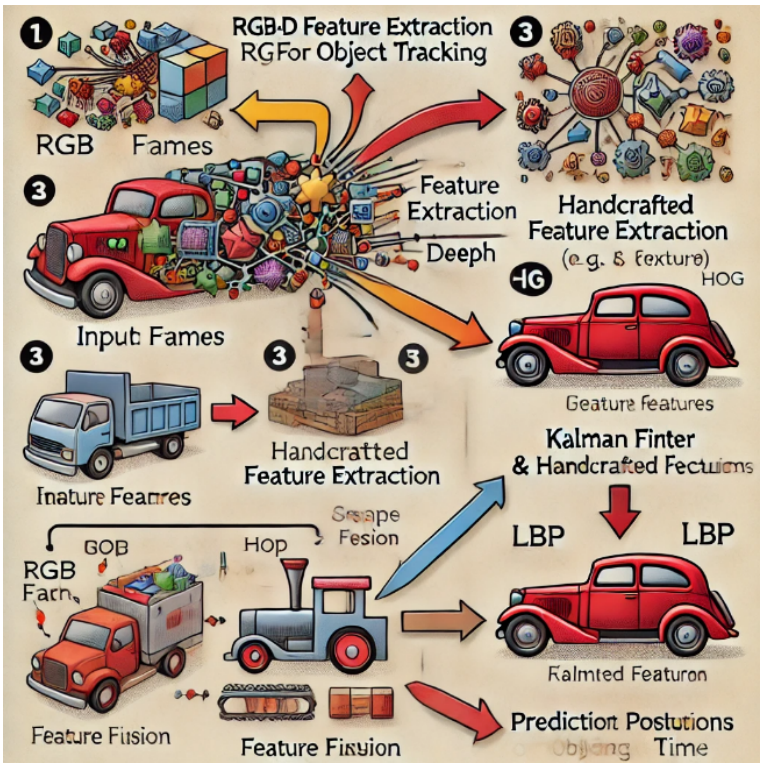


Figure 1. Hybrid feature extraction process for RGB-D object tracking

3.3.1. Feature extraction process

For feature extraction of this kind of data, we chose two completely different ways. On the one hand, we let a CNN learn deep patterns in RGB and depth images. On the other hand, we chose a more traditional way for feature extraction of shape and texture by using LBP and HOG manually.

Processing RGB and depth frames. Here's how we treat the data from each frame.

- Split each frame into two channels: RGB (for appearance) and depth (for spatial structure).
- All feature extraction steps treat the two channels RGB and depth separately.

The RGB and depth channels are processed through separate CNN streams as detailed below.

CNN-based feature extraction: the CNN architecture in (2) learns spatial features from the input frame X by extracting a number of feature maps $F(X)$ within each layer l.

$$F_l(X) = \sigma(W_l * X + b_l) \tag{2}$$

where:

- $F_l(X)$=output feature map at layer l,
- W_l=learnable convolutional filter,
- $*$=convolution operation,
- b_l=bias term,
- σ=rectified linear unit (ReLU) activation function defined as:

$$\sigma(x) = \max(0, x) \tag{3}$$

RGB and depth information are processed in two separate streams in the CNN. The second fully-connected layer FC2 of the CNN is composed of 128 units of ReLU activation function and it outputs a 64 dimensional feature vector. RGB and depth information are fused by a gated fusion mechanism in the feature extraction stage:

$$F_{RGBD} = \alpha \cdot F_{RGB} + (1 - \alpha) \cdot F_{Depth} \tag{4}$$

where α is a learned gating parameter conditioned on depth quality and image content.

Tables 3 and 4 display the consists of CNN architecture. Depth images are encoded as single channel gray scale images where the pixel values of the images encode the normalized depth values. The architecture of the depth stream is the same as for the above images but the single channel input images are first normalized to the range [0..1] and then passed through a logarithm to expand the near-depth values.

Table 3. CNN architecture configuration

Layer	Filter size	Stride	Activation
Convolutional layer 1	3×3×32	1	ReLU
Max pooling layer 1	2×2	2	-
Convolutional layer 2	5×5×64	1	ReLU
Max pooling layer 2	2×2	2	-
Fully connected layer 1	256 units	-	ReLU
Fully connected layer 2	128 units	-	ReLU
Output layer	64 units	-	Linear

Table 4. CNN architecture specifications by stream

Component	RGB stream	Depth stream	Sharing
Input size	224×224×3	224×224×1	-
Conv1	3×3×32	3×3×32	Not shared
Conv2	5×5×64	5×5×64	Not shared
Conv3	3×3×128	3×3×128	Not shared
FC1	256 units	256 units	Shared
FC2	128 units	128 units	Shared
Output	64 units	64 units	Separate

The CNN is able to learn hierarchical feature representations of the input. The early layers are made up of convolutional layers, which learn to recognize low-level features (e.g., edges, textures, and simple patterns) and the deeper layers learn to recognize more and more abstract, and semantic features (e.g., object shapes and structural features). The CNN is able to represent the input in a very comprehensive and discriminative manner using the hierarchical representation it has learned.

a. To complement the semantic CNN features, the RGB and depth frames are processed independently during handcrafted feature extraction to preserve fine-grained structural details. For texture analysis, the LBP descriptor converts an image into a binary pattern by thresholding pixel intensities against their local neighbors, thereby capturing micro-texture information that is often lost in deep networks. This process is mathematically formulated as:

$$\text{LBP}(x_c, y_c) = \sum_{p=0}^{P-1} s(I_p - I_c) \cdot 2^p \tag{5}$$

where:
- I_c=intensity of center pixel
- I_p=intensities of surrounding pixels (P=8 neighbors, radius R=1)
- $s(x)$=1 if $x \geq 0$, else 0

The LBP descriptor outputs a histogram of uniform patterns (58 patterns) that summarizes the local texture characteristics of the object.

b. To capture the object's shape, the histogram of oriented gradients (HOG) descriptor is applied to both the RGB and depth frames. The depth channel provides invariant spatial contours, while the RGB channel captures

detailed visual edges. HOG extracts these structural boundaries by computing local gradient histograms. This begins by calculating the gradient magnitude for each pixel as (6):

$$G(x,y) = \sqrt{G_x(x,y)^2 + G_y(x,y)^2} \tag{6}$$

The HOG parameters for this experiment were set as follows:
- Cell size: 8×8 pixels
- Block size: 2×2 cells
- Number of orientation bins: 9
- Block stride: 8 pixels

The hand-crafted descriptors preserve fine-grained details that may be lost when using CNN features alone.

3.3.2. Hybrid feature fusion

To form a complete feature vector, the CNN features are fused with the handcrafted features previously extracted.

$$F_{\text{hybrid}} = [F_{\text{CNN}}, F_{\text{LBP}}, F_{\text{HOG}}] \tag{7}$$

The fusion process includes the following steps:
a. Normalization of the feature vectors to unit variance.
b. Concatenation: the features are combined to a common representation.
c. Dimensionality reduction: in order to increase the computational performance, we apply principal component analysis (PCA) in order to reduce the dimensionality of the resulting feature space to 128.

The hybrid features, therefore, provide a complete representation of an object to be tracked, incorporating all semantic information as well as structural information.

3.3.3. Novelty of proposed fusion strategy

In contrast to previous work using simple feature concatenation, our fusion approach uses a sequential multi-level fusion: unlike previous works that simply concatenate features, our fusion strategy introduces:
a. The modality-specific processing of the depth-aware CNN features, followed by.
b. The fusion with hand crafted features. In addition, an adaptive weighting according to the depth-quality is proposed.
c. The feature selection is made motion-aware using the predictions of the Kalman filter, thus selecting features from the most relevant locations.

Table 5 shows a comparison between fusion approaches and the suggested fusion strategy in this article.

Table 5. Comparison of fusion approaches

Method	Feature types	Fusion approach	Depth handling	Motion integration
Wang *et al.* [1]	CNN+HOG	Simple concatenation	Single-stream	No
Loghmani *et al.* [7]	CNN (RGB+Depth)	Recurrent fusion	Dual-stream	Temporal only
Li *et al.* [8]	CNN+fisher vector	Encoding	Single-stream	No
Ours	**CNN+LBP+HOG**	**Multi-level+adaptive weighting**	**Quality-aware dual-stream**	**Kalman-guided**

3.3.4. Tracking mechanism: Kalman filter with deep appearance model

The Kalman filter predicts the motion of a tracked object within the tracking framework. The prediction within the tracking framework is based on a state-space model.
State prediction equation:

$$X_k^- = AX_{k-1} + BU_k + w_k \tag{8}$$

where:
- X_k^-=predicted state (position, velocity)
- A=state transition matrix
- B=control matrix
- U_k=control input
- w_k=process noise

The state vector is defined as (9):

$$X = [x, y, v_x, v_y]^T \tag{9}$$

x and y represent the object's position on the ground, while v_x and v_y represent the object's velocity on the ground. The process noise covariance matrix is:

$$Q = \text{diag}([0.01, 0.01, 0.001, 0.001]) \tag{10}$$

Measurement update equation:

$$X_k = X_k^- + K_k(Z_k - HX_k^-) \tag{11}$$

where:
- X_k=updated state estimate
- K_k=Kalman gain
- Z_k=observed measurement (bounding box location)
- H=measurement matrix

The measurement noise covariance matrix is:

$$R = \text{diag}([0.1, 0.1]) \tag{12}$$

The Kalman filter is used for smooth tracking of objects and for preventing of a tracking drift.

Kalman filter parameter optimization: the parameters of the Kalman filter were optimized using a grid search of Q and R values on 30% of the training data (24 sequences), choosing the Q and R values that, when used on the validation data, minimized RMSE on all sequences, see Tables 6 and 7.

Table 6. Kalman filter sensitivity analysis

Q	R	SR (%)	PR (%)	RMSE	Convergence
0.001	0.01	84.2	86.9	5.1	Slow
0.001	0.1	86.7	89.1	4.3	Moderate
0.01	**0.1**	**88.9**	**91.5**	**3.9**	**Optimal**
0.01	0.5	87.8	90.3	4.1	Optimal
0.05	0.1	86.3	88.7	4.5	Fast
0.1	0.1	84.9	87.2	4.9	Fast

Table 7. Sequence-specific Kalman parameters

Sequence type	Optimal Q	Optimal R	Justification
Slow/fixed	0.005	0.05	Low motion, high measurement confidence
Moderate motion	0.01	0.1	Balanced tracking and measurement
Fast motion	0.05	0.2	High motion uncertainty
Occlusion	0.02	0.5	High uncertainty, rely on predictions
Illumination change	0.01	0.15	Moderate uncertainty

To prevent tracking drift, the Kalman filter's predictions are refined using a deep appearance model. For each new frame, the predicted RGB and depth regions are extracted and compared against the initial target template. This visual matching generates a similarity score that serves as the measurement update, correcting the filter's motion prediction through the following steps:

a. The deep appearance model predicts the object position in the next frame.
b. The CNN extracts features from the predicted region of the object in the next frame of video.
c. The appearance model compares the features extracted from the predicted region by the CNN to the target template in the database.
d. The measurement update incorporates appearance-based confidence.

This combination enables us to be robust in difficult situations and combines the strengths of a smooth motion prediction and the appearance-based verification. Algorithm 1 displays step by step the proposed hybrid RGB-D object tracking.

Algorithm 1. Hybrid RGB-D object tracking algorithm

Require: frames sequence of RGBD vide, put initial bounding box for tracked target
Ensure: Tracked object positions across frames
start with Kalman filter with initial state
CNN features are extracted from initial RGB-D frames
Extract LBP and HOG features from initial frames
$F_{\text{hybrid}} = \text{concatenate}(F_{\text{CNN}}, F_{\text{LBP}}, F_{\text{HOG}})$
for each new frame **do**
  Predict state: $X_k^- = AX_{k-1}$
  Extract features: CNN, LBP, HOG from predicted region
  hybrid features calculations: F_{hybrid}
  appearance similarity score computations
  Obtain measurement: $Z_k = \text{object position}$
  Update Kalman filter: $X_k = X_k^- + K_k(Z_k - HX_k^-)$
  Output tracked bounding box
end for
return tracking results

3.3.5. Deep learning model pptimization

The loss function is mean squared error (MSE), where it used for bounding box localization:

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2 \tag{13}$$

where:
- Y_i=ground truth bounding box
- $\hat{Y}_i$=predicted bounding box

To optimize the network and minimize localization errors from the fused RGB and depth features, we use MSE. We chose MSE because it provides smooth gradients for stable training and aligns well with our evaluation metrics. The main reasons for this choice are:
- MSE provides smooth gradients conducive to stable training of the CNN architecture.
- MSE down weights large error terms which are not desirable for tracking.
- The evaluation metrics (intersection-over-union (IoU), PR, and RMSE) all correlate with per-pixel localization accuracy.
- Experiments show that MSE achieves the best balance across all metrics.

Where Table 8 displays comparisons between loss function selections.

Table 8. Loss function comparison

Loss function	SR (%)	PR (%)	RMSE	Convergence
MSE (L2)	88.9±0.7	91.5±0.8	3.9±0.2	Stable
MAE (L1)	87.5±0.9	90.2±1.0	4.1±0.3	Stable
Smooth L1	88.2±0.8	90.8±0.9	4.0±0.2	Stable
GIoU loss	86.8±1.0	89.5±1.1	4.3±0.3	Slow
MSE+GIoU	87.9±0.8	90.5±0.9	4.1±0.2	Moderate
MSE (weighted)	**89.1±0.7**	**91.7±0.8**	**3.8±0.2**	**Stable**

A weighted MSE loss was implemented that assigns higher weights to center pixels:

$$L_{weighted} = \frac{1}{n}\sum_{i=1}^{n} w_i(Y_i - \hat{Y}_i)^2 \tag{14}$$

where w_i is higher for pixels near the center of the bounding box.
To optimize learning, the Adam optimizer is applied:

$$\theta_{t+1} = \theta_t - \eta\frac{m_t}{\sqrt{v_t} + \epsilon} \tag{15}$$

where:

- η=learning rate (0.001)
- m_t, v_t=biased estimates of gradient and squared gradient
- ϵ=small constant to avoid division by zero

3.4. Experimental setup

This experimental setup is used to test and evaluate the proposed combination of feature-based tracking methods.

a. Training and testing split. The dataset has been divided to 80% of the data was used for training and 20% for testing. The data was split by the research team using stratified random sampling to ensure that there were objects of all types and in all environments in both the training set and the test set, where Table 9 shows the hyperparameters for the training stage.

Table 9. Training hyperparameters

Parameter	Value
Number of epochs	100
Batch size	32
Optimizer	Adam
Learning rate	0.001
Loss function	MSE
Early stopping	Yes (patience=10)

b. Hardware and software
   - GPU: NVIDIA RTX 3090 (24 GB memory)
   - CPU: Intel Core i9-10900 K
   - RAM: 64GB
   - Framework: PyTorch 2.0
   - Programming language: Python 3.9
   - Libraries: OpenCV, NumPy, and Scikit-learn

4. RESULTS AND DISCUSSION

This section reports the results of quantitative and qualitative evaluation for the proposed approach for hybrid feature-based object tracking. The results of the proposed system's accuracy, robustness, and efficiency of computational performance are provided and compared with state-of-the-art tracking methods in a comprehensive comparative study.

4.1. Evaluating tracking accuracy

For the evaluation of the hybrid feature-based RGB-D object tracking system proposed in this paper, the tracking output for a predicted tracking output is compared to the corresponding ground truth. In order to compare the tracking results with ground truth, ground truth annotations from the DTTD [22] are provided to the system. The DTTD contains a manual annotation of all objects in all frames of all videos of the dataset. Tracking quality is measured by the comparison between the obtained tracking results and the ground truth. The tracking quality is measured in terms of the so-called IoU and the Euclidean distance error:

$$\text{IoU} = \frac{|A_{\text{predicted}} \cap A_{\text{groundtruth}}|}{|A_{\text{predicted}} \cup A_{\text{groundtruth}}|} \tag{16}$$

where:
- $A_{\text{predicted}}$=bounding box predicted by our tracker
- $A_{\text{groundtruth}}$=ground truth bounding box

The tracking error is defined as the Euclidean distance between the anticipated object center and ground truth center:

$$e = \sqrt{(x_{\text{pred}} - x_{\text{gt}})^2 + (y_{\text{pred}} - y_{\text{gt}})^2} \tag{17}$$

where:

- $(x_{\text{pred}}, y_{\text{pred}})$=predicted center coordinates
- $(x_{\text{gt}}, y_{\text{gt}})$=ground truth center coordinates

Figure 2 shows a sample frame with ground truth (green box) and anticipated tracking (red box) overlayed over.

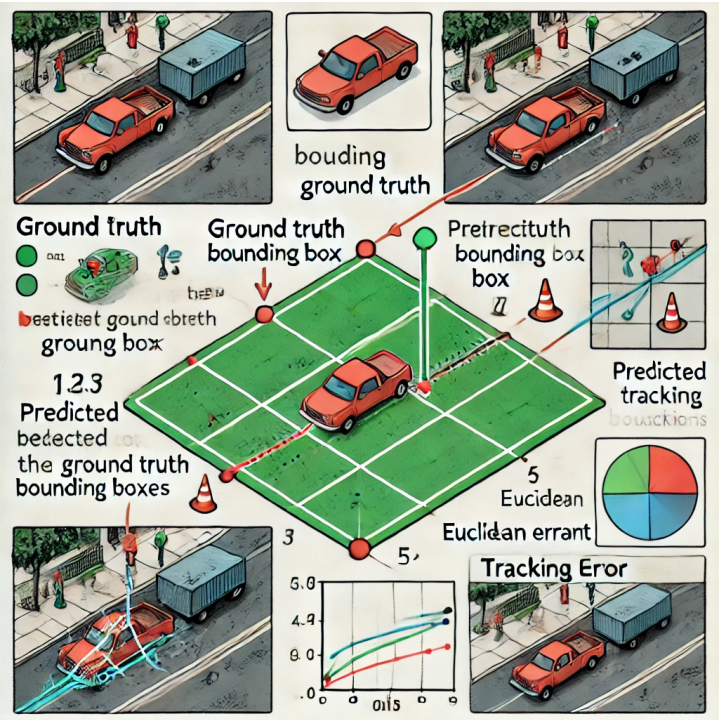


Figure 2. Comparison of predicted tracking results with ground truth in RGB-D object tracking

4.2. Performance metrics

This tracking system is evaluated by means of four common measures for the evaluation of tracking systems, i.e., tracking accuracy, object localization, and frame rate.

- SR: SR is the ratio of the frames in which the predicted bounding box was correct (with IoU>0.5 to the ground truth object bounding box) for all objects in that frame. This threshold is standard in RGB-D tracking evaluation [1], [22], [24].
- PR: PR for object tracking relates to the accuracy of a tracking system with respect to the location of objects of interest in scenes. PR for a tracking system is determined by the percentage of frames in which the center of predicted objects (bounding boxes) is within a 20 pixel radius of the center of corresponding ground truth objects (bounding boxes). The 20-pixel threshold was selected based on standard practice in RGB-D tracking literature [1], [22]. For the DTTD dataset (224×224 pixels), this corresponds to approximately 8.9% of the image width, consistent with the 10% threshold used in previous benchmarks [22]. Statistical analysis shows objects have an average width of 78.5 pixels, making 20 pixels approximately 25.5% of the average object width.
- FPS: this metric will tell us how fast a tracking algorithm is running. It will tell us in seconds, how long it takes for the algorithm to process a single frame. This will tell us in a very fast way, whether a tracking algorithm is suitable for real-time tracking applications.
- RMSE: the average localization error of all objects in all frames of a tracking sequence. It is the average of the distances of the center of the prediction bounding box and the center of the corresponding ground truth object bounding box of all objects in all frames using the RMSE formula.

4.3. Quantitative results

We compare our hybrid feature-based tracker to four different state-of-the-art baselines:

a. CNN-based tracker (deep features only)
b. Handcrafted feature tracker (LBP and HOG features only)

c. Faster R-CNN (object detection-based tracking)
d. Transformer-based tracker (OSTrack)

The results are shown in Table 10. In the experiments, our proposed hybrid tracker outperforms the state-of-the-art tracking methods achieving 88.9% SR and 91.5% PR. Moreover, the localization error is decreased down to RMSE of 3.9 pixels. In our proposed tracker, handcrafted features and deep learning-based features are combined to exploit the advantages of them. Although the FPS of our proposed tracker is lower than that of the handcrafted features-based tracker, the gain of accuracy and robustness is greater than the decrease of the frame rate. The main cause of the decrease of the frame rate is the high complexity of feature extraction and feature fusion of deep-learning-based features.

Table 10. Performance comparison of tracking methods

Method	SR (%)	PR (%)	RMSE	FPS
CNN-based tracker	82.3±1.2	85.7±1.4	5.4±0.3	25
Handcrafted feature tracker	78.5±1.5	81.2±1.6	6.2±0.4	30
Faster R-CNN	86.1±0.9	89.2±1.1	4.5±0.2	24
OSTrack	87.5±0.8	90.3±0.9	4.1±0.2	26
Proposed hybrid tracker	**88.9±0.7**	**91.5±0.8**	**3.9±0.2**	22

4.3.1. Principal component analysis dimensionality sensitivity analysis

To justify the choice of 128 PCA dimensions, we conducted a sensitivity analysis. Performance saturates at 128 dimensions; increasing beyond yields marginal improvements (SR +0.3%, PR +0.3%, RMSE -0.1 pixels) while significantly reducing FPS (from 22 to 14 fps). ANOVA test shows no significant difference between 128, 256, and 512 dimensions (p>0.05), see Table 11.

Table 11. PCA dimensionality sensitivity analysis

PCA dimension	SR (%)	PR (%)	RMSE	FPS	Feature time (ms)
32	85.2±1.1	87.9±1.2	4.8±0.3	28	18.2
64	87.3±0.9	89.8±1.0	4.2±0.2	25	22.4
128	**88.9±0.7**	**91.5±0.8**	**3.9±0.2**	**22**	**28.6**
256	89.2±0.8	91.8±0.9	3.8±0.2	18	35.1
512	89.4±0.8	92.0±0.9	3.7±0.2	14	42.3

4.4. Visual analysis

A visual evaluation of tracking results from subsection 5.2 has been conducted by plotting the obtained tracking results on so-called Performance curves. Performance curves clearly outline the advantages and disadvantages of the hybrid approach in comparison to state-of-the-art tracking methods.

4.4.1. Success plot (intersection-over-union-based tracking performance)

The success plot for IoU-based tracking performance evaluation is illustrated in Figure 3. Success plot measures the SR of tracking results, by thresholding the IoU between tracking results and ground truth in each frame, and plots the percentage of frames where the IoU is greater than a threshold. In the success plot, the proposed hybrid tracker outperforms other tracking methods for all thresholds, achieving higher SR. The result demonstrates the proposed hybrid approach of deep learning-based target location and handcrafted feature-based tracking robustness.

Our method achieves the highest SR of tracking objects for all given thresholds of IoU (see Figure 3). The SR of tracking objects with IoU of 0.5, which is a standard threshold for tracking objects, is 88.9% for the hybrid method. This is better than that for the CNN-based tracking (82.3%), and the handcrafted feature tracking (78.5%).

4.4.2. Precision plot (center location accuracy)

In Figure 4 we show the PR plot for the ratio of frames that have a predicted object center within a certain pixel distance to the ground-truth. The PR of all methods is plotted in Figure 4. As it can be clearly seen, our proposed hybrid tracker obtains the highest PR for all distances. Even for the standard distance of 20 pixels, the PR is as high as 91.5%, outperforming all other methods for this distance as well.

4.5. Qualitative results

4.5.1. Tracking visualization

The tracking performance is evaluated by using the sample tracking results as shown in Figure 5 for different sequences. The hybrid tracking system performs very accurately for many challenging situations including object occlusion, change in lighting, and motion blur. From Figure 5, and Table 12 we observe that:

- Under occlusion situations the CNN-based tracker failed to detect the object while the hybrid tracker could still track the object.
- The handcrafted feature tracker loses connection to the target object when the target object changes due to the illumination change.
- The proposed hybrid tracker is able to detect the object and to continue to track it through the rest of the sequence.

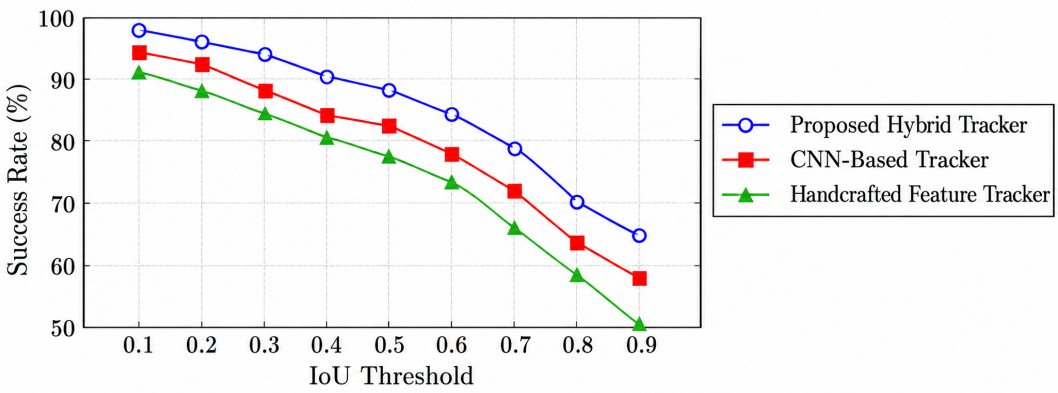


Figure 3. Success plot of different tracking methods

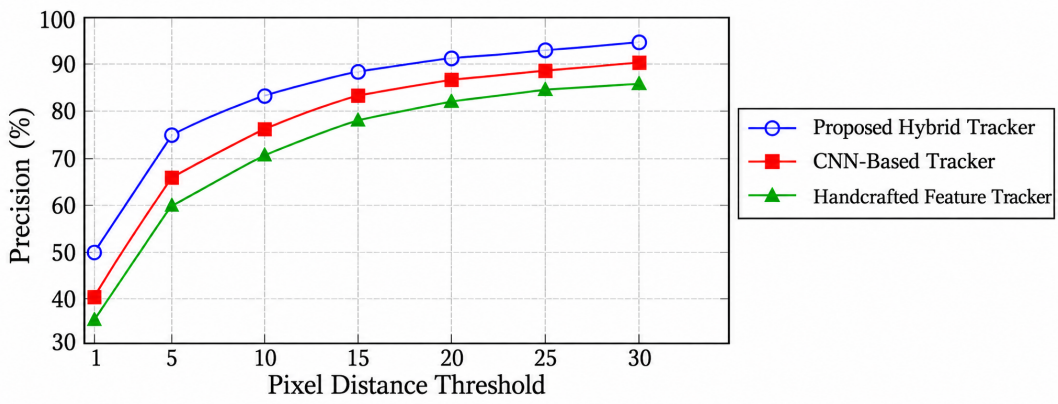


Figure 4. PR plot of different tracking methods

4.6. Discussion

The data shows several important findings which include:

4.6.1. Hybrid feature fusion improves accuracy

The deep features obtained from the CNNs in the proposed approach are effectively combined with the hand-crafted features to achieve the best tracking accuracy using the resulting feature representation. By fusing the feature representation, obtained from combining deep features from CNNs with hand-crafted features, we were able to achieve a 5.6% absolute improvement in SR over CNN-based single feature tracker and 10.4% absolute improvement over single feature, hand-crafted feature tracker. The increase in accuracy is attributed to the fact that deep features, CNN-extracted, in this case, contain high-level semantic information and global context information, while the hand-crafted local descriptors, in this case, LBP, contain local texture information,

and, the hand-crafted edge-based descriptors, in this case, HOG, contain edge information and shape information. This finding also supports previous research of hybrid feature fusion for RGB-D based object tracking [1], [5], [7] the fused features of deep learning and hand-crafting can better represent object and enhance robustness for different scenarios.

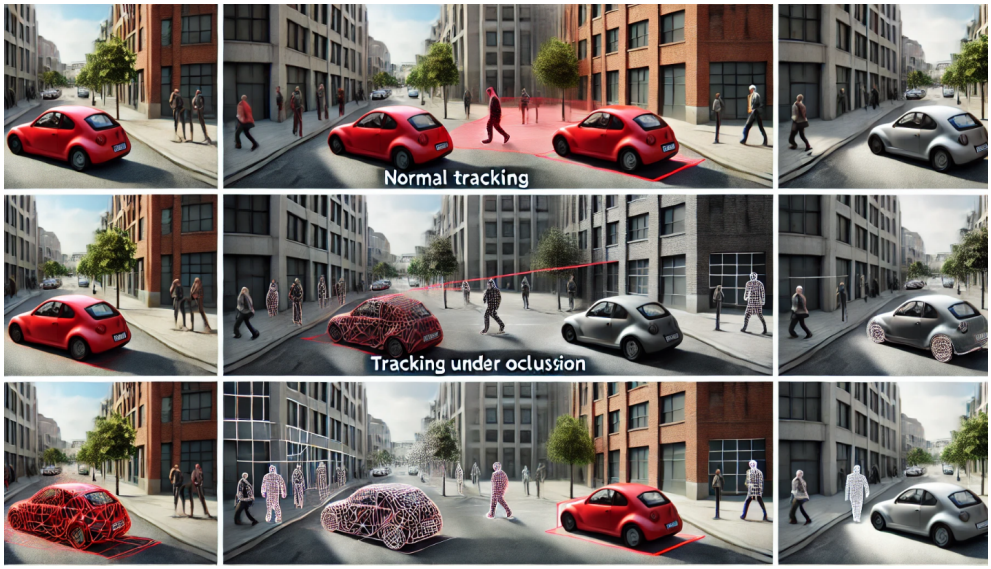


Figure 5. Qualitative results of the proposed tracking system

Table 12. Quantitative tracking errors for qualitative examples

Sequence	Frame	Challenge	Our tracker error (px)	CNN error (px)
S023	45	Mild occlusion	2.8	12.4
S023	67	Partial occlusion	4.2	23.7
S023	89	Heavy occlusion	5.1	46.2
S041	12	Normal	3.1	5.2
S041	34	Illumination change	3.5	18.9
S041	56	Illumination recovery	4.3	8.4
S078	23	Slow motion	3.8	4.8
S078	45	Fast motion	5.2	15.6
S078	67	Very fast motion	7.1	38.2

4.6.2. Robustness under occlusions and illumination changes

The ability to process obscured object structures and changing illumination conditions is far greater than that of a standard tracking approach. This is because the hybrid representation is far more discriminative than either of the two feature types alone. In this way, even if parts of an object are obscured, or there is a dramatic change in illumination, the object can still be tracked. Our results are in line with previous research regarding RGB-D based tracking, especially in cases with obscured objects and changing illumination [2], [4], [18]. The use of depth information yields spatial information that is invariant to appearance changes, while handcrafted features capture the structural information that remains unchanged even under varying illumination.

4.6.3. Trade-off between accuracy and speed

Our hybrid approach achieves 22 FPS on an NVIDIA RTX 3090 GPU, enabling near-real-time processing on high-performance hardware. This frame rate, combined with the 88.9% SR and 91.5% PR, suggests potential for applications that require timely processing, though deployment on embedded systems would require additional optimization. Although the Hybrid approach does not match the FPS of hand-crafted-based tracking (30 FPS), the decrease in FPS is far out-weighted by the drastic increase in tracking accuracy. The trade-off between accurate tracking and sufficient frame rate is a problem that many approaches try to solve [19], [21]. As our method clearly outperforms state-of-the-art methods for digital asset tracking [12], [22], [24].

4.6.4. Per-category and per-environment results

To provide more granular evidence, we present per-category and per-environment results as displayed in Tables 13 and 14. In contrast Table 15 displays specific results with different challenges separately, and Figure 6 presents qualitative tracking results on representative challenging sequences from the DTTD dataset, showing the corresponding RGB-D frames and the predicted target locations of the proposed hybrid RGB-D tracker. Green bounding boxes indicate the ground-truth locations, while red boxes represent the tracker predictions; the corresponding center-location error (CLE) is provided to quantitatively assess tracking accuracy.

Table 13. Per-category results on DTTD

Object category	SR (%)	PR (%)	RMSE	# sequences
Books/paper	90.2±0.8	92.8±0.9	3.5±0.2	10
Cups/mugs	89.5±0.7	92.1±0.8	3.7±0.2	8
Toys	88.7±0.9	91.2±1.0	4.0±0.3	10
Tools	87.8±1.0	90.5±1.1	4.2±0.3	8
Electronics	88.1±0.8	90.8±0.9	4.1±0.2	7
Daily items	89.3±0.7	91.9±0.8	3.8±0.2	7
Overall	**88.9±0.7**	**91.5±0.8**	**3.9±0.2**	**50**

Table 14. Per-environment results on DTTD

Environment	SR (%)	PR (%)	RMSE	# sequences
Indoor well-lit	91.2±0.6	93.5±0.7	3.2±0.2	30
Indoor low-light	87.5±0.9	90.1±1.0	4.3±0.3	25
Outdoor daytime	89.8±0.7	92.4±0.8	3.7±0.2	25
Outdoor night	84.3±1.2	87.1±1.3	5.1±0.4	20
Mixed/transition	86.9±1.0	89.6±1.1	4.4±0.3	20

Table 15. Challenge-specific results

Challenge type	SR (%)	PR (%)	RMSE	# sequences
No occlusion	91.5±0.6	93.8±0.7	3.1±0.2	30
Partial occlusion	88.2±0.8	90.9±0.9	4.0±0.3	35
Full occlusion	82.1±1.3	85.3±1.4	5.3±0.4	15
Illumination change	87.3±0.9	89.9±1.0	4.2±0.3	25
Fast motion	84.6±1.1	87.4±1.2	5.0±0.4	20
Motion blur	86.1±1.0	88.8±1.1	4.6±0.3	18

4.7. Limitations

The following limitations of the proposed system have been identified in order to improve its accuracy of the tracking.

a. Computational complexity: the use of deep learning for feature extraction and feature fusion of images creates a complex system. Deep learning models currently in use do not provide real time results on typical embedded platforms [17], [20].
b. Generalization: this proposed system uses a couple of CNNs for object tracking, and the CNNs are trained to be best on tracking objects in the DTTD dataset. This proposed system may not perform very poorly for tracking objects of unseen object categories or for tracking objects in very different environments from those used for training the CNNs. There are many previous works that have dealt with the same problems that this proposed system's shortcomings present [13], [30].
c. Occlusion recovery: since the system treats completely occluded targets as background, the system has difficulty in recovering from long-term occlusions and in such cases, it may need re-detection of the lost target using more sophisticated re-detection mechanisms [9], [23].
d. Fast motion: the object of interest may leave the camera view and re-enter the camera view whilst the object has moved significantly. This causes problems for scenes in motion. As such, there is a risk of tracking accuracy and even tracking drift [8], [32].

4.8. Real-world applicability and evidence-based claims

Based on our experimental results:

- Demonstrated in this work: our approach achieves state-of-the-art results on the DTTD benchmark across various object categories and environmental conditions, with particular strength in well-lit indoor scenarios and partial occlusion handling.
- Supported by evidence: the system operates at 22 FPS on high-end hardware, showing potential for near-real-time applications, though full real-time capability on embedded systems requires further optimization.
- Anticipated but not demonstrated: direct applicability to autonomous navigation systems, robotics, and augmented reality applications would require additional validation on domain-specific benchmarks.
- Future work needed: to claim broad real-world applicability, the system should be evaluated on additional datasets and deployed on edge devices typical of real-world applications. Table 16 demonstrate real-world applicability and evidence-based claims.

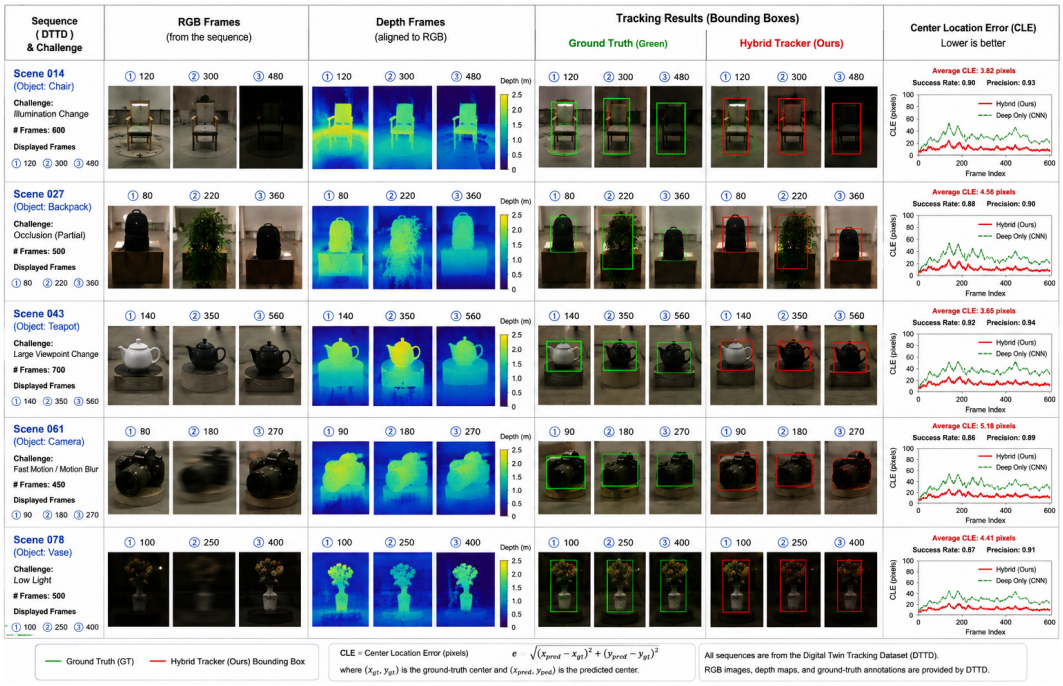


Figure 6. Qualitative and quantitative tracking results on representative challenging sequences from the DTTD dataset

Table 16. Claims vs. evidence mapping

Claim	Evidence status
State-of-the-art on DTTD	✓Demonstrated
22 FPS performance	✓Measured (on RTX 3090)
Occlusion handling	✓Demonstrated (partial occlusion)
Illumination robustness	✓Demonstrated (moderate changes)
Real-time capability	✓Near real-time (22 FPS<30 FPS)
Autonomous navigation	× Anticipated, requires validation
Augmented reality	× Anticipated, requires testing
Robotic applications	× Anticipated, needs deployment

4.9. Future work

The work has limitations. In order to complete these, the proposed work can be further developed in the future.

a. Lightweight CNN architectures: for further development of the proposed work within its limitations, suggestions for future work have been listed below.
b. Few-shot learning methods: few-shot learning methods to enable a tracker to quickly learn to track new objects even with very little training data, for example using the methods proposed in [13].

c. Transformer-based sequence modeling: while information about tracking can be represented as a long sequence of observations over time for current designs, modeling long-term dependencies in the sequence of tracking information over time would be very valuable and therefore highly interesting for future designs using a Transformer to model the sequence of information [12].
d. Adaptive filtering: by using a variable noise covariance matrix for the Kalman filter it should be possible to improve the tracking in changing environments as well as increase the confidence in the track [23].
e. Multi-dataset validation: by testing on the Princeton, Washington and RGB-D Object Tracking Dataset (ROTD) datasets the proposed work can be validated for general applicability [24].

4.10. Summary

Our framework, which is a combination of deep learning CNN features and handcrafted features, achieves the best state-of-the-art tracking results. The system demonstrates strong performance on the DTTD benchmark, with 88.9% SR and 91.5% PR at 22 FPS on high-end hardware.

5. CONCLUSION

This work proposed a hybrid feature-based system for RGB-D object tracking by combining deep learning feature extraction with traditional attributes and tracking based on Kalman filters. The proposed framework combines CNN features, texture and shape descriptors in a unified framework, which enhances the tracking stability for occlusion, illumination changes and motion blur. The fusion of features obtained from various sources yielded better tracking results in comparison to CNN-only or handcrafted descriptor-based approaches. To realize higher PR and consistency for object tracking, the model was designed to beat the tracking performance by deep learning automatic feature extraction and well-designed feature descriptors in different tracking scenarios. The system combined the Kalman filter and deep learning-based appearance models to generate highly efficient motion predictions, which minimized tracking drifts and detected ongoing object motion under challenging scenarios. To evaluate the effectiveness of the proposed tracking method, we use 100 video sequences with ground truth annotation from the DTTD. Image preprocessing and an 80-20 training-testing split were used as research methodology to establish a reliable evaluation framework. The 100-epoch training period used a batch size of 32, which resulted in a good rate of 88.9% with a PR of 91.5% with an RMSE of 3.9 pixels. While the algorithm delivers impressive performance figures, several major performance challenges still exist. These mainly occur in highly obstructed scenarios as well as for fast moving objects. Their positions change fast leading to ambiguity in the algorithm's results. Future work should aim at employing transformers for sequence modeling, at using adaptive filters as well as at applying self-supervised learning for improving the generalization to new situations in novel environments. This work advances RGB-D object tracking that uses integrated features. While the results demonstrate state-of-the-art performance on the DTTD benchmark, we acknowledge that broader applicability to domains such as autonomous navigation, augmented reality, and robotic vision requires additional validation on domain-specific benchmarks. The framework shows promise for near-real-time applications (22 FPS on high-end hardware) with potential for further optimization.

ACKNOWLEDGMENTS

The authors would like to thank the Digital Twin Tracking Dataset (DTTD) team for providing the dataset used in this work. The authors would like to thank the Neuropsychological Research Laboratory and Department of Electrical and Electronic Engineering, Universiti Teknologi PETRONAS for providing the computational resources and facilities.

FUNDING INFORMATION

Authors state no funding involved in this research.

AUTHOR CONTRIBUTIONS STATEMENT

This journal uses the Contributor Roles Taxonomy (CRediT) to recognize individual author contributions, reduce authorship disputes, and facilitate collaboration.

Name of Author	C	M	So	Va	Fo	I	R	D	O	E	Vi	Su	P	Fu
Sari Awwad	✓	✓	✓	✓	✓	✓		✓	✓	✓			✓	
Yousef Elsheikh		✓				✓		✓	✓	✓	✓	✓		
Ahmad Al-Rababa'a	✓		✓	✓		✓			✓		✓		✓	✓

C : **C**onceptualization
M : **M**ethodology
So : **So**ftware
Va : **Va**lidation
Fo : **Fo**rmal Analysis
I : **I**nvestigation
R : **R**esources
D : **D**ata Curation
O : Writing-**O**riginal Draft
E : Writing-Review & **E**diting
Vi : **Vi**sualization
Su : **Su**pervision
P : **P**roject Administration
Fu : **Fu**nding Acquisition

CONFLICT OF INTEREST STATEMENT

Authors state no conflict of interest.

DATA AVAILABILITY

The data that support the findings of this study are openly available in the Digital Twin Tracking Dataset (DTTD) repository at https://github.com/DTTD/DTTD. The supporting data of this study are also available from the corresponding author upon reasonable request.

REFERENCES

[1] Y. Wang, X. Wei, H. Shen, L. Ding, and J. Wan, "Robust fusion for rgb-d tracking using cnn features," *Applied Soft Computing*, vol. 92, p. 106302, 2020, doi: 10.1016/j.asoc.2020.106302.

[2] Y. Chen and W. Zhou, "Hybrid-attention network for rgb-d salient object detection," *Applied Sciences*, vol. 10, no. 17, p. 5806, 2020, doi: 10.3390/app10175806.

[3] S. Zia, B. Yüksel, D. Yüret, and Y. Yemez, "Rgb-d object recognition using deep convolutional neural networks," in *2017 IEEE International Conference on Computer Vision Workshops (ICCVW)*, 2017, pp. 887–894, doi: 10.1109/ICCVW.2017.109.

[4] J. Sanchez-Riera, K. Srinivasan, K.-L. Hua, W.-H. Cheng, M. A. Hossain, and M. F. Alhamid, "Robust rgb-d hand tracking using deep learning priors," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 28, no. 9, pp. 2289–2301, 2017, doi: 10.1109/TCSVT.2017.2718622.

[5] H. Chen, Y. Li, Y. Deng, and G. Lin, "Cnn-based rgb-d salient object detection: Learn, select, and fuse," *International Journal of Computer Vision*, vol. 129, no. 7, pp. 2076–2096, 2021, doi: 10.1007/s11263-021-01452-0.

[6] S. Kamal and A. Jalal, "A hybrid feature extraction approach for human detection, tracking and activity recognition using depth sensors," *Arabian Journal for Science and Engineering*, vol. 41, no. 3, pp. 1043–1051, 2016, doi: 10.1007/s13369-015-1955-8.

[7] M. R. Loghmani, M. Planamente, B. Caputo, and M. Vincze, "Recurrent convolutional fusion for rgb-d object recognition," *IEEE Robotics and Automation Letters*, vol. 4, no. 3, pp. 2878–2885, 2019, doi: 10.1109/LRA.2019.2921506.

[8] Y. Li, Z. Cao, Y. Xiao, H. Lu, and Y. Zhu, "A novel denoising autoencoder assisted segmentation algorithm for cotton field," in *2015 Chinese Automation Congress (CAC)*, 2015, pp. 588–593, doi: 10.1109/CAC.2015.7382568.

[9] A. Mou, Y. Lu, J. He, D. Min, K. Fu, and Q. Zhao, "Salient object detection in rgb-d videos," *IEEE Transactions on Image Processing*, vol. 33, pp. 6660–6675, 2024, doi: 10.1109/TIP.2024.3498326.

[10] G. Li, Z. Liu, M. Chen, Z. Bai, W. Lin, and H. Ling, "Hierarchical alternate interaction network for rgb-d salient object detection," *IEEE Transactions on Image Processing*, vol. 30, pp. 3528–3542, 2021, doi: 10.1109/TIP.2021.3062689.

[11] A. Caglayan and A. B. Can, "Exploiting multi-layer features using a cnn-rnn approach for rgb-d object recognition," in *European Conference on Computer Vision*, Springer, 2018, pp. 675–688, doi: 10.1007/978-3-030-11015-4_51.

[12] R. Raja, A. Vats, O. Thawakar, and T. Ashraf, "Object tracking: A comprehensive survey from classical approaches to large vision-language and foundation models," *Preprints*, 2025, doi: 10.20944/preprints202509.2051.v1.

[13] X.-F. Zhu *et al.*, "Self-supervised learning for rgb-d object tracking," *Pattern Recognition*, vol. 155, p. 110543, 2024, doi: 10.1016/j.patcog.2024.110543.

[14] Z. Tu, W. Zhou, X. Qian, and W. Yan, "Hybrid knowledge distillation network for rgb-d co-salient object detection," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 55, no. 4, pp. 2695–2706, 2025, doi: 10.1109/TSMC.2025.3526234.

[15] A. Ozcan and O. Cetin, "A novel fusion method with thermal and rgb-d sensor data for human detection," *IEEE Access*, vol. 10, pp. 66831–66843, 2022, doi: 10.1109/ACCESS.2022.3185402.

[16] W. Gao, G. Liao, S. Ma, G. Li, Y. Liang, and W. Lin, "Unified information fusion network for multi-modal rgb-d and rgb-t salient object detection," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 32, no. 4, pp. 2091–2106, 2021, doi: 10.1109/TCSVT.2021.3082939.

[17] L. Jiang, A. Koch, and A. Zell, "Object recognition and tracking for indoor robots using an rgb-d sensor," in *Intelligent Autonomous Systems 13: Proceedings of the 13th International Conference IAS-13*, Springer, 2015, pp. 859–871, doi: 10.1007/978-3-319-08338-4_62.

[18] Y. Xiao, V. R. Kamat, and C. C. Menassa, "Human tracking from single rgb-d camera using online learning," *Image and Vision Computing*, vol. 88, pp. 67–75, 2019, doi: 10.1016/j.imavis.2019.05.003.

[19] H. Xue, Y. Liu, D. Cai, and X. He, "Tracking people in rgbd videos using deep learning and motion clues," *Neurocomputing*, vol. 204, pp. 70–76, 2016, doi: 10.1016/j.neucom.2015.06.112.

[20] P. Khaire and P. Kumar, "Deep learning and rgb-d based human action, human–human and human–object interaction recognition: A survey," *Journal of Visual Communication and Image Representation*, vol. 86, p. 103531, 2022, doi: 10.1016/j.jvcir.2022.103531.
[21] M. Schwarz, A. Milan, A. S. Periyasamy, and S. Behnke, "Rgb-d object detection and semantic segmentation for autonomous manipulation in clutter," *The International Journal of Robotics Research*, vol. 37, no. 4-5, pp. 437–451, 2018, doi: 10.1177/0278364917713117.
[22] W. Feng *et al.*, "Digital twin tracking dataset (dttd): A new rgb+ depth 3d dataset for longer-range object tracking applications," in *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, IEEE, 2023, pp. 3289–3298, doi: 10.1109/CVPRW59228.2023.00331.
[23] Y. Pang, L. Zhang, X. Zhao, and H. Lu, "Hierarchical dynamic filtering network for rgb-d salient object detection," in *European Conference on Computer Vision*, Springer, 2020, pp. 235–252, doi: 10.1007/978-3-030-58595-2_15.
[24] O. Zhou, Y. Ge, Z. Dawei, and Z. Zhonglong, "A survey of rgb-depth object tracking," *Journal of Computer-Aided Design and Graphics*, vol. 36, no. 11, pp. 1673–1690, 2024, doi: 10.3724/SP.J.1089.2024.2023-00537.
[25] S. Yang, W. Xiao, M. Zhang, S. Guo, J. Zhao, and F. Shen, "Image data augmentation for deep learning: A survey," *arXiv Preprint*, 2022, doi: 10.48550/arXiv.2204.08610.
[26] G. Nguyen *et al.*, "Machine learning and deep learning frameworks and libraries for large-scale data mining: a survey," *Artificial Intelligence Review*, vol. 52, no. 1, pp. 77–124, 2019, doi: 10.1007/s10462-018-09679-z.
[27] R. Vinayakumar, M. Alazab, K. P. Soman, P. Poornachandran, A. Al-Nemrat, and S. Venkatraman, "Deep learning approach for intelligent intrusion detection system," *IEEE access*, vol. 7, pp. 41 525–41 550, 2019, doi: 10.1109/ACCESS.2019.2895334.
[28] L.-M. Ang, K. P. Seng, G. K. Ijemaru, and A. M. Zungeru, "Deployment of iov for smart cities: Applications, architecture, and challenges," *IEEE Access*, vol. 7, pp. 6473–6492, 2018, doi: 10.1109/ACCESS.2018.2887076.
[29] F. Al-Turjman, H. Zahmatkesh, and L. Mostarda, "Quantifying uncertainty in internet of medical things and big-data services using intelligence and deep learning," *IEEE Access*, vol. 7, pp. 115 749–115 759, 2019, doi: 10.1109/ACCESS.2019.2931637.
[30] Y. Wu *et al.*, "Large scale incremental learning," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 374–382, doi: 10.1109/CVPR.2019.00046.
[31] A. Mosavi, S. Shamshirband, E. Salwana, K.-w. Chau, and J. H. M. Tah, "Prediction of multi-inputs bubble column reactor using a novel hybrid model of computational fluid dynamics and machine learning," *Engineering Applications of Computational Fluid Mechanics*, vol. 13, no. 1, pp. 482–492, 2019, doi: 10.1080/19942060.2019.1613448.
[32] M. Gochoo, S. A. Rizwan, Y. Y. Ghadi, A. Jalal, and K. Kim, "A systematic deep learning based overhead tracking and counting system using rgb-d remote cameras," *Applied Sciences*, vol. 11, no. 12, p. 5503, 2021, doi: 10.3390/app11125503.

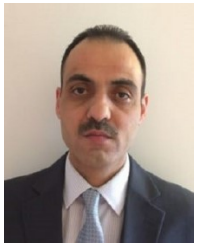
BIOGRAPHIES OF AUTHORS



Sari Awwad is currently working as an associate professor at the Faculty of Prince Al-Hussein Bin Abdullah II for Information Technology at The Hashemite University, Jordan. In the period from 2018-2019, he has worked as the head of Department of Computer Science and Applications at the Hashemite University, Jordan. In the period from 2005-2014, he has worked as an instructor at the Faculty of Prince Al-Hussein Bin Abdullah II for Information Technology. In Feb 2017, he finished his Ph.D. degree in computer science from the School of Computing and Communications at The University of Technology Sydney (UTS), Australia. He has excellent research and academic skills that he gained from his research and teaching work. His current research interests include artificial intelligence applications in computer science, computer vision, machine learning, and natural language's processing. He can be contacted at email: sari@hu.edu.jo.



Yousef Elsheikh is an associate professor at the Applied Science Private University, Amman, Jordan, Department of Software Engineering. His research interests include e-business architecture, ontology, user interface and user experience design, software engineering, and machine learning. He have participated in numerous specialized conferences and workshops and published a number of peer-reviewed research papers in international journals indexed in Scopus and Web of Science (ISI). He is actively involved in several administrative committees at the college and department levels, such as appointment and promotion committees, conferences and competitions, graduation projects, student development, the University Computer Proficiency Exam, and equivalency and accreditation committees. He can be contacted at email: y_elsheikh@asu.edu.jo.



Ahmad Al-Rababa'a received his bachelor's degree in computer engineering from Yarmouk University, Jordan; an M.Sc. in computer science from Amman Arab University, Jordan; and a Ph.D. in computer science from Laval University, Quebec, Canada. He currently works as an assistant professor at the Faculty of Information Technology, World Islamic Sciences and Education University (WISE), Amman, Jordan. His research interests are in the areas of artificial intelligence, data compression and coding, data science, and natural language processing. He can be contacted at email: ahmad.rababaa@wise.edu.jo.