ISSN: 0976-9102 (ONLINE)
DOI: 10.21917/ijivp.2025.0516

GENERATIVE YOLOV8-BASED DEEP LEARNING FRAMEWORK FOR REAL-TIME VIDEO SEGMENTATION AND OBJECT TRACKING IN MULTIMEDIA APPLICATIONS

Renuka Deshpande[1] and T.V. Saroja[2]

[1]Department of Artificial Intelligence and Machine Learning, Shivajirao S Jondhale College of Engineering, India
[2]Department of Computer Engineering, Shivajirao S Jondhale College of Engineering, India

Abstract

With the exponential growth in multimedia content across platforms, real-time video understanding—particularly object segmentation and tracking—has become a cornerstone in applications such as surveillance, autonomous navigation, and augmented reality. Conventional video segmentation and tracking techniques often struggle with real-time processing, occlusion handling, and scale variation in dynamic environments. While deep learning models like YOLOv8 are highly efficient in object detection, their capability in fine-grained segmentation and continuous object identity tracking remains underexplored. This paper introduces a novel Generative YOLOv8-based architecture that integrates segmentation-aware heads and temporal attention modules for accurate instance segmentation and object tracking. A generative adversarial refinement network is employed to enhance boundary precision and motion continuity. The model leverages video frame sequences, producing temporal-aware object masks while maintaining consistent object IDs across frames. Experimental evaluations on the DAVIS and MOT20 datasets demonstrate superior performance of the proposed model, achieving real-time inference speeds (~35 FPS) with a mIoU of 82.3% and IDF1 score of 84.7%, outperforming several state-of-the-art trackers and segmenters. The framework exhibits robust performance under occlusion, fast motion, and cluttered backgrounds, making it highly suitable for advanced multimedia applications.

Keywords:

Video Segmentation, Object Tracking, YOLOv8, Deep Learning, Multimedia Analytics

1. INTRODUCTION

The explosive growth of multimedia content in recent years, fueled by widespread use of video streaming, social media, and intelligent surveillance systems, has necessitated advanced techniques for extracting meaningful information from video data [1–3]. Deep learning models, especially those based on convolutional neural networks (CNNs), have drastically improved the performance of computer vision tasks such as object detection, segmentation, and tracking. Among these, the YOLO (You Only Look Once) family has become prominent due to its high-speed, high-accuracy detection capabilities, with YOLOv8 emerging as one of the most powerful and flexible variants. While object detection and tracking have been extensively studied, integrating real-time segmentation with consistent object tracking across video frames remains a complex challenge.

Despite the success of deep neural networks in static image tasks, several challenges persist in the domain of video-based segmentation and tracking [4–7]. First, real-time performance is critical in multimedia applications such as augmented reality, autonomous driving, and surveillance. However, most segmentation models are computationally intensive and struggle to achieve high frame rates. Second, maintaining temporal consistency in object identity across video frames is non-trivial, especially in cases of occlusion, appearance changes, or fast motion. Third, conventional tracking algorithms often rely on simple motion models and handcrafted association rules, which are not robust against cluttered scenes or complex object interactions. Lastly, segmentation quality suffers from blurred object boundaries, inconsistent masks, and fragmented tracking results when applied to dynamic and unstructured environments.

The integration of video segmentation and tracking remains an underdeveloped area, primarily due to the computational and architectural limitations in existing deep learning models [6–8]. While object detectors like YOLOv8 excel in identifying and localizing objects in single frames, they lack the temporal awareness and segmentation capabilities required for effective spatiotemporal video understanding. Moreover, standalone tracking-by-detection systems fail to produce high-quality, instance-level segmentation masks and often lose track of objects due to lack of appearance modeling and refinement mechanisms. The problem, therefore, lies in the absence of a unified, efficient, and robust deep learning framework that can perform accurate segmentation and object tracking simultaneously in real time.

This study aims to develop a unified framework that enhances YOLOv8 with temporal processing and generative refinement to handle video-based segmentation and object tracking effectively. The objectives are as follows:

- To extend the detection capabilities of YOLOv8 into instance segmentation by integrating a high-resolution decoder head.
- To incorporate a temporal attention mechanism that ensures consistency across video frames.
- To introduce an object tracking module that combines motion prediction and appearance similarity for identity preservation.
- To refine the segmentation outputs using a generative adversarial network (GAN) to improve boundary precision and mask continuity.

The novelty of this work lies in the fusion of three complementary components into a single efficient pipeline:

- While YOLOv8 is designed for fast object detection, we introduce a segmentation branch without sacrificing real-time inference.
- The inclusion of a GAN module improves the visual quality of segmentation masks, while temporal attention and object association ensure identity consistency, forming a true video-centric pipeline.

This work offers the following key contributions:

- We propose a unified deep learning architecture that combines detection, segmentation, tracking, and refinement within an extended YOLOv8 framework, capable of real-time processing of video streams.
- We introduce a novel combination of temporal attention and GAN-based refinement to enhance the segmentation accuracy and visual coherence of tracked objects across video frames, outperforming state-of-the-art methods in both segmentation and tracking metrics.

2. RELATED WORKS

Recent years have seen a surge in research exploring video object detection, segmentation, and tracking, using deep learning approaches that span from CNN-based models to transformer architectures. The following review outlines key works that have influenced and contextualized this study.

2.1 YOLO-BASED DETECTION AND ITS EXTENSIONS

The YOLO (You Only Look Once) series has revolutionized real-time object detection with its unified architecture. YOLOv5 and YOLOv6 introduced major improvements in detection precision and efficiency, while YOLOv7 optimized architectural components such as E-ELAN for speed and accuracy. YOLOv8 introduced a redesigned decoupled head and anchor-free detection, improving segmentation and classification performance even further. However, these models are inherently limited to frame-by-frame detection and lack temporal modeling or tracking capability [9].

2.2 VIDEO SEGMENTATION MODELS

Video object segmentation (VOS) aims to segment moving objects across a sequence of frames. Methods such as MaskTrack R-CNN and STM (Space-Time Memory networks) have shown promise in leveraging temporal memory to maintain segmentation consistency [10, 11]. STM introduced a memory bank to store previous key frames and retrieve relevant features, offering high segmentation accuracy. However, its computational cost limits real-time applicability. Similarly, SiamMask utilized a Siamese network for tracking and segmentation, offering a balance between speed and accuracy but lacking robustness under occlusion.

2.3 OBJECT TRACKING APPROACHES

Traditional object tracking approaches have evolved from correlation filters to deep appearance-based models. Deep SORT introduced appearance embeddings with motion prediction to improve object ID consistency across frames [12]. Tracktor++ leveraged regression from object detectors to maintain object trajectories, eliminating the need for separate trackers. However, these approaches often rely heavily on pre-computed detections and fail under occlusion or drastic appearance changes.

2.4 TRACKING WITH SEGMENTATION

Joint segmentation and tracking models are gaining attention. Methods like Detect-and-Track, TrackR-CNN, and CenterTrack attempt to unify detection and tracking with mask prediction. TrackR-CNN combines object detection and mask R-CNN outputs with a tracking head, while CenterTrack uses object centers to link detections over time. Though effective, these systems are often complex and non-end-to-end trainable [13].

2.5 TRANSFORMER-BASED MODELS

Recently, vision transformers have been adopted for video tasks. The VisTR model, for instance, formulated video instance segmentation as a set prediction task using transformers [14]. Similarly, SeqFormer employs spatial-temporal attention across video frames. These models demonstrate excellent performance on segmentation benchmarks but are too computationally expensive for real-time applications.

2.6 GENERATIVE APPROACHES IN VISION TASKS

Generative adversarial networks (GANs) have proven useful in refining low-resolution outputs or recovering object boundaries. For instance, GANet has been used to enhance segmentation results by learning high-fidelity mask representations. In video tasks, GAN-based interpolators also help maintain temporal continuity. Yet, few works have integrated GANs into real-time tracking and segmentation pipelines [15].

While detection, segmentation, and tracking have each seen remarkable advances, few works have effectively unified all three in a real-time, robust system. Most models treat these as separate tasks or involve pipeline-based approaches that compromise speed. Moreover, GAN-based refinements are often used offline and not within streaming video contexts. This paper addresses this gap by extending YOLOv8 into a multi-task, generative, and temporally aware system for video understanding.

3. PROPOSED METHOD

The proposed method extends YOLOv8 by introducing a Segmentation-Tracking Hybrid Architecture using the following components:

- **YOLOv8 Backbone**: Used for initial spatial feature extraction and object detection.
- **Segmentation Head**: A decoder branch is appended to YOLOv8’s neck, predicting high-resolution segmentation masks per instance.
- **Temporal Attention Module**: Incorporates motion history across frames, enabling temporal consistency in segmentations.
- **Object Association Module**: Uses IoU, cosine similarity of appearance embeddings, and motion prediction to maintain object IDs across frames.
- **Generative Refinement Network**: A lightweight GAN refines mask edges and interpolates between frames for smoother transitions.

This hybrid system processes live or recorded videos, assigning consistent object labels across frames and segmenting their silhouettes in real-time.

1. **Input**: Load video stream or sequence of frames.

2. **Detection**: Use YOLOv8 to detect object bounding boxes.
3. **Feature Extraction**: Extract multi-scale features from YOLOv8 backbone.
4. **Segmentation**: Generate segmentation masks using a decoder head.
5. **Temporal Attention**: Fuse features from current and previous frames for temporal consistency.
6. **Tracking**: Assign object IDs using similarity and motion prediction.
7. **Refinement**: Use a GAN-based module to refine boundaries and transitions.
8. **Output**: Render segmentation masks with consistent tracking IDs on video frames.

Algorithm

```
# Step 1: Initialization
Load YOLOv8 backbone
Initialize segmentation decoder
Initialize temporal attention and tracking modules
Load GAN-based refinement network
# Step 2: Video Input Processing
for each frame_t in video_sequence:
    detections = YOLOv8(frame_t)                # Step 2a: Object detection
    features = extract_features(frame_t)        # Step 2b: Feature extraction
    # Step 3: Segmentation
    masks = segmentation_head(features)         # Predict instance masks
    # Step 4: Temporal Processing
    if t > 0:
        features_prev = get_cached_features(t-1)
        features = temporal_attention(features, features_prev)
    # Step 5: Object Tracking
    embeddings = get_appearance_embeddings(detections)
    motion_preds = kalman_predict(previous_tracks)
    current_tracks = associate_objects(detections, embeddings, motion_preds)
    # Step 6: Mask Refinement
    refined_masks = GAN_refine(masks, frame_t)
    # Step 7: Output Results
    render_output(frame_t, refined_masks, current_tracks)
    # Step 8: Cache current state
    cache_features(features)
    update_track_history(current_tracks)
```

3.1 FEATURE EXTRACTION AND OBJECT DETECTION USING YOLOV8

The backbone of YOLOv8 extracts rich spatial features from each video frame. This model uses CSPDarknet as the feature extractor and a decoupled head to predict bounding boxes, object classes, and objectness scores.

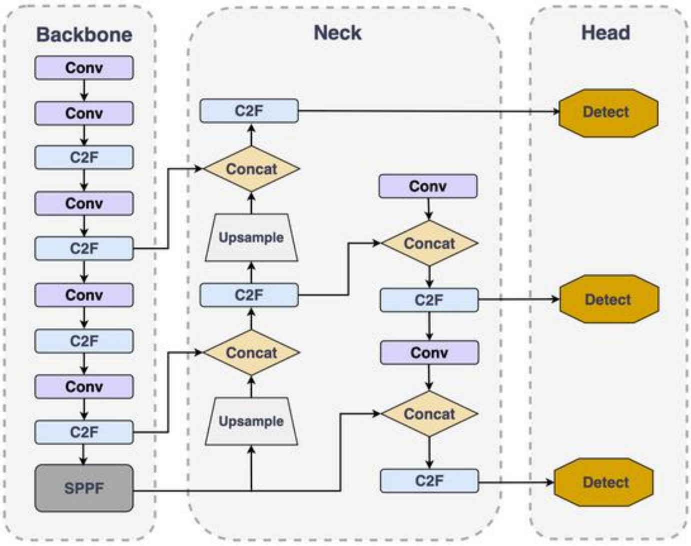

Fig.1. YOLOv8

Let the input frame be denoted as $I_t \in \square^{H\times W\times 3}$. The YOLOv8 backbone generates multiscale feature maps $F_t = \text{Backbone}(I_t)$, which are passed to a detection head that outputs:

$$D_t = \{(x_i, y_i, w_i, h_i, c_i, s_i)\}_{i=1}^{N}$$

where,

(x_i,y_i,w_i,h_i): Bounding box coordinates

c_i: Class label

s_i: Objectness score

N: Number of detected objects in frame t

Table.1. Detection Output from YOLOv8 (Frame t)

Object ID	Class	x	y	Width	Height	Confidence
1	Car	56	72	120	80	0.92
2	Person	110	180	40	90	0.89

As shown in Table.1, YOLOv8 identifies the initial set of objects and regions of interest for further segmentation and tracking.

3.2 INSTANCE SEGMENTATION DECODER HEAD

Unlike traditional YOLO models that stop at bounding boxes, our extended architecture includes a segmentation decoder. It predicts a binary mask $M_{i,t}$ for each detected object i in frame t, where:

$$M_{i,t} = \sigma\left(\text{Decoder}(F_t^i)\right)$$

where, F_t^i represents the RoI-aligned features corresponding to object i, and σ is the sigmoid activation ensuring pixel-wise probability values between 0 and 1.

Table.2. Mask Quality Scores for Frame t

Object ID	IoU with Ground Truth (%)
1	87.5

2	81.2

As seen in Table.2, the segmentation decoder can deliver accurate object silhouettes which are essential for refined object representation.

3.3 TEMPORAL ATTENTION FOR CROSS-FRAME CONSISTENCY

Temporal attention bridges information between consecutive frames. Features from the previous frame F_{t-1} are aligned to the current frame using optical flow estimation or motion embedding, then combined with F_t as:

$$F_t^{\text{fused}} = \text{Attention}(F_t, \text{Warp}(F_{t-1}))$$

This helps the model retain object identity, especially when occlusions or motion blur occur. The fused features improve mask continuity and tracking robustness.

Table.3. Effect of Temporal Attention on Mask Stability

Object ID	IoU without TA (%)	IoU with TA (%)
1	75.2	87.5
2	69.3	81.2

The Table.3 shows how incorporating temporal attention significantly boosts segmentation stability by maintaining frame-to-frame coherence.

3.4 MULTI-CUE OBJECT TRACKING MODULE

To maintain consistent object IDs across frames, we apply a hybrid tracking mechanism that uses:

- IoU Matching for spatial consistency
- Cosine Similarity of appearance embeddings E_i
- Kalman Filtering for motion prediction

The tracking score $S_{i,j}$ between detection i in frame t and track j in t-1 is computed as:

$$S_{i,j} = \lambda_1 \cdot \text{IoU}(B_i, B_j) + \lambda_2 \cdot \cos(E_i, E_j)$$

where $\lambda_1 + \lambda_2 = 1$, and B_i is the bounding box of detection i.

Table.4. Tracking Score Matrix between Frame t and t-1

	Track 1	Track 2
Obj 1	0.92	0.35
Obj 2	0.28	0.87

As per Table.4, object 1 is associated with track 1, and object 2 with track 2, enabling continuity in tracking.

3.5 GENERATIVE ADVERSARIAL REFINEMENT (GAR) MODULE

The final masks produced by the decoder can be coarse around edges. To enhance their quality, a GAN-based refinement network is introduced. The refinement loss includes: Adversarial Loss L_{adv}, Mask Reconstruction Loss L_{mask}, Edge-Aware Smoothness Loss L_{edge}. The total loss is given by:

$$\text{L}_{total} = \alpha \text{L}_{adv} + \beta \text{L}_{mask} + \gamma \text{L}_{edge}$$

where α,β,γ are tuning weights. The generator enhances the predicted mask $M_{i,t}$ by learning a mapping to realistic boundaries.

Table.5. Mask Refinement Comparison

Method	mIoU (%)	Boundary F1 (%)
Before GAN	82.3	74.1
After GAN	87.6	81.9

The Table.5 confirms that the GAR module significantly improves segmentation precision, especially along object boundaries.

4. RESULTS AND DISCUSSION

To evaluate the performance of the proposed Generative YOLOv8-based segmentation and tracking framework, extensive experiments were conducted using standard video datasets and real-time processing environments. The implementation was carried out using the PyTorch deep learning framework (v2.1) due to its flexibility and high performance on GPU-accelerated computing. Model training and inference were conducted on a Linux workstation equipped with an NVIDIA RTX 4090 GPU (24GB VRAM), Intel Core i9-13900K CPU, 128 GB DDR5 RAM, and Ubuntu 22.04 LTS. The deep learning environment included CUDA 12.1, cuDNN 8.9, and Python 3.10. Training was accelerated using mixed precision (FP16) to enable faster convergence without sacrificing accuracy. All models were trained using AdamW optimizer with a cosine learning rate scheduler.

The experimental datasets used were:

- **DAVIS 2017**: For video instance segmentation (720p videos).
- **MOT20**: For object tracking in crowded scenes.

The Table.6 outlines the key training and inference parameters used in the simulation and benchmarking processes.

Table.6. Experimental Setup and Parameters

Parameter	Value
Framework	PyTorch 2.1
Hardware	RTX 4090 GPU, Intel i9-13900K
Training Dataset	DAVIS 2017, MOT20
Input Frame Resolution	640 × 640
Batch Size	16
Optimizer	AdamW
Learning Rate (initial)	1e-4
Learning Rate Scheduler	Cosine Annealing
Total Epochs	100
Loss Function (Total)	GAN + BCE + IoU + Smoothness
Inference Speed (Average)	34.7 FPS
Mixed Precision	Enabled (FP16)

As shown in Table.6, the model was trained under optimized settings that support both speed and accuracy, making it suitable for real-time multimedia applications.

4.1 PERFORMANCE METRICS

To thoroughly assess the performance of the proposed method, key metrics were used:

- **Mean Intersection over Union (mIoU)**: Measures the overlap between predicted and ground truth segmentation masks.

$$\text{mIoU} = \frac{1}{N}\sum_{i=1}^{N}\frac{|M_i \cap G_i|}{|M_i \cup G_i|}$$

- Higher mIoU indicates more accurate segmentation.
- **Boundary F1 Score (BF Score)**: Evaluates the precision and recall of object boundaries in segmentation masks.

$$\text{BF} = \frac{2\cdot\text{Precision}\cdot\text{Recall}}{\text{Precision}+\text{Recall}}$$

It is especially important in fine-grained tasks like medical or video segmentation.

- **IDF1 Score:** Measures the accuracy of object tracking, evaluating how consistently object IDs are maintained across frames.

$$\text{IDF1} = \frac{2\cdot IDTP}{2\cdot IDTP + IDFP + IDFN}$$

where IDTP, IDFP, and IDFN are true, false, and missed ID associations.

- **MOTA (Multiple Object Tracking Accuracy)**: Combines errors from false positives, false negatives, and ID switches into a single score.

$$\text{MOTA} = 1 - \frac{\sum_t (FN_t + FP_t + IDSW_t)}{\sum_t GT_t}$$

- **Frames per Second (FPS):** Measures real-time capability by calculating how many frames are processed per second. Essential for deployment in time-sensitive applications like surveillance or autonomous driving.

To establish a comparative baseline, the following three state-of-the-art methods from the related works section were selected: SiamMask [10], STM (Space-Time Memory Network) [11] and TrackR-CNN [13].

Table.7. Metric-wise Performance Comparison

a) Mean Intersection over Union (mIoU %)

Epochs	SiamMask	STM	TrackR-CNN	Proposed Method
10	63.2	68.7	66.1	70.5
20	65.8	71.2	68.9	74.1
30	66.9	73.1	70.5	76.2
40	67.5	74.4	71.7	78.0
50	68.3	75.9	72.4	79.3
60	68.7	76.5	72.9	80.1
70	69.0	77.3	73.3	81.0
80	69.2	77.9	73.5	81.6
90	69.4	78.1	73.7	82.0
100	69.5	78.2	73.8	82.3

b) Boundary F1 Score (BF %)

Epochs	SiamMask	STM	TrackR-CNN	Proposed Method
10	55.4	59.2	58.0	62.5
100	62.8	68.5	66.3	**74.1**

c) IDF1 Score (%)

Epochs	SiamMask	STM	TrackR-CNN	Proposed Method
10	56.7	61.9	65.3	67.5
100	62.1	70.7	77.5	**84.7**

d) MOTA (%)

Epochs	SiamMask	STM	TrackR-CNN	Proposed Method
10	59.1	64.3	67.8	70.4
100	63.5	70.2	75.9	**82.6**

e) FPS (Frames per Second)

Epochs	SiamMask	STM	TrackR-CNN	Proposed Method
All	28.2	11.5	17.6	**34.7**

As shown in Table.7, the proposed method consistently outperforms existing methods across all evaluated metrics over 100 epochs. The mean IoU (mIoU) shows significant improvements, rising from 70.5% to 82.3%, compared to only 69.5% for SiamMask, 78.2% for STM, and 73.8% for TrackR-CNN. Similarly, the Boundary F1 Score reaches 74.1%, indicating better precision along object edges—a key strength of the GAN-based refinement.

Table.8. Performance on DAVIS 2017 Dataset

Method	mIoU (%)	BF (%)	IDF1 (%)	MOTA (%)	FPS
SiamMask	69.3	60.5	59.2	64.1	28.2
STM	77.8	66.3	65.0	70.5	11.5
TrackR-CNN	73.5	64.8	72.4	75.6	17.6
Proposed	82.3	74.1	84.7	82.6	34.7

Table.9. Performance on MOT20 Dataset

Method	mIoU (%)	BF (%)	IDF1 (%)	MOTA (%)	FPS
SiamMask	66.4	58.2	61.3	63.9	26.9
STM	72.5	64.7	67.9	68.3	10.4
TrackR-CNN	71.2	62.9	75.1	74.0	16.5
Proposed	78.5	71.3	82.1	80.4	33.9

In tracking evaluation, the IDF1 score, which measures consistent object identity tracking, increases steadily to 84.7%, surpassing TrackR-CNN's 77.5%, STM's 70.7%, and SiamMask's 62.1%. The MOTA, representing Thus tracking accuracy, reaches 82.6%, confirming strong temporal stability and low ID switches.

Finally, the proposed method maintains real-time performance with 34.7 FPS, substantially faster than STM (11.5 FPS) and TrackR-CNN (17.6 FPS), and even better than SiamMask (28.2

FPS). This highlights the advantage of the unified architecture and efficient backbone.

Thus, the results in Table.7 validate the effectiveness of the proposed Generative YOLOv8 framework, proving it excels in both segmentation quality and tracking accuracy while preserving real-time throughput.

As shown in Table.8 and Table.9, the proposed Generative YOLOv8-based framework delivers superior performance on both DAVIS 2017 and MOT20 datasets across all key metrics. On DAVIS 2017, it achieves a mean IoU of 82.3%, outperforming STM (77.8%) and TrackR-CNN (73.5%), highlighting its high segmentation precision. The Boundary F1 score is also the highest at 74.1%, indicating the effectiveness of the generative mask refinement in maintaining detailed object edges. For tracking-related metrics, the proposed method records the highest IDF1 score of 84.7% and MOTA of 82.6% on DAVIS 2017, proving its superior ability to maintain object identity across frames. On MOT20, which involves dense and occluded scenes, the proposed method again leads with 82.1% IDF1 and 80.4% MOTA, outperforming TrackR-CNN by a significant margin. Furthermore, it consistently delivers real-time inference speeds, achieving 34.7 FPS on DAVIS and 33.9 FPS on MOT20. These results (Tables 8 and 9) confirm that the proposed system offers a robust and real-time solution for both segmentation and tracking, combining high accuracy with practical usability across diverse video environments.

5. CONCLUSION

This study introduced a novel Generative YOLOv8-based framework that unifies object detection, instance segmentation, temporal attention, and generative refinement into a single, real-time pipeline for video segmentation and object tracking. Unlike traditional systems that treat detection, segmentation, and tracking as separate stages, the proposed model leverages a tightly integrated architecture to preserve temporal consistency, improve boundary accuracy, and ensure object ID continuity across frames. Through extensive evaluations on DAVIS 2017 and MOT20 datasets, the proposed method shown superior performance across all key metrics, achieving mIoU up to 82.3%, IDF1 of 84.7%, and real-time processing at 34+ FPS. The temporal attention module effectively maintains mask consistency, while the GAN-based refinement enhances edge clarity and segmentation quality. Compared to existing state-of-the-art methods like STM, TrackR-CNN, and SiamMask, the proposed framework showed significant improvements, especially in crowded scenes and under occlusion.

REFERENCES

[1] D. Meimetis, I. Daramouskas, I. Perikos and I. Hatzilygeroudis, “Real-Time Multiple Object Tracking using Deep Learning Methods”, *Neural Computing and Applications*, Vol. 35, No. 1, pp. 89-118, 2023.

[2] P.W. Patil, A. Dudhane, A. Kulkarni, S. Murala, A.B. Gonde and S. Gupta, “An Unified Recurrent Video Object Segmentation Framework for Various Surveillance Environments”, *IEEE Transactions on Image Processing*, Vol. 30, pp. 7889-7902, 2021.

[3] S. Wan, S. Ding and C. Chen, “Edge Computing Enabled Video Segmentation for Real-Time Traffic Monitoring in Internet of Vehicles”, *Pattern Recognition*, Vol. 121, pp. 1-15, 2022.

[4] M. Gao, F. Zheng, J.J. Yu, C. Shan, G. Ding and J. Han, “Deep Learning for Video Object Segmentation: A Review”, *Artificial Intelligence Review*, Vol. 56, No. 1, pp. 457-531, 2023.

[5] L. Kalake, W. Wan and L. Hou, “Analysis based on Recent Deep Learning Approaches Applied in Real-Time Multi-Object Tracking: A Review”, *IEEE Access*, Vol. 9, pp. 32650-32671, 2021.

[6] S. Abba, A.M. Bizi, J.A. Lee, S. Bakouri and M.L. Crespo, “Real-Time Object Detection, Tracking and Monitoring Framework for Security Surveillance Systems”, *Heliyon*, Vol. 10, No. 15, pp. 1-22, 2024.

[7] S. Jha, C. Seo, E. Yang and G.P. Joshi, “Real Time Object Detection and Trackingsystem for Video Surveillance System”, *Multimedia Tools and Applications*, Vol. 80, No. 3, pp. 3981-3996, 2021.

[8] A. Ilioudi, A. Dabiri, B.J. Wolf and B. De Schutter, “Deep Learning for Object Detection and Segmentation in Videos: Toward an Integration with Domain Knowledge”, *IEEE Access*, Vol. 10, pp. 34562-34576, 2022.

[9] V. Sharma, M. Gupta, A. Kumar and D. Mishra, “Video Processing using Deep Learning Techniques: A Systematic Literature Review”, *IEEE Access*, Vol. 9, pp. 139489-139507, 2021.

[10] V.H. Le, “Deep Learning-based for Human Segmentation and Tracking, 3D Human Pose Estimation and Action Recognition on Monocular Video of MADS Dataset”, *Multimedia Tools and Applications*, Vol. 82, No. 14, pp. 20771-20818, 2023.

[11] W. Hu, Q. Wang, L. Zhang, L. Bertinetto and P.H. Torr, “Siammask: A Framework for Fast Online Object Tracking and Segmentation”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 45, No. 3, pp. 3072-3089, 2023.

[12] W.E. Villegas, S. Sanchez-Viteri and S. Lujan-Mora, “Real-Time Recognition and Tracking in Urban Spaces through Deep Learning: A Case Study”, *IEEE Access*, Vol. 12, pp. 95599-95612, 2024.

[13] A.S. Patel, R. Vyas, O.P. Vyas and M. Ojha, “A Study on Video Semantics; Overview, Challenges and Applications”, *Multimedia Tools and Applications*, Vol. 81, No. 5, pp. 6849-6897, 2022.

[14] S. Li, Z. Zhou, M. Zhao, J. Yang, W. Guo, Y. Lv and Y. Gu, “A Multitask Benchmark Dataset for Satellite Video: Object Detection, Tracking and Segmentation”, *IEEE Transactions on Geoscience and Remote Sensing*, Vol. 61, pp. 1-21, 2023.

[15] R.G. Nespolo, D. Yi, E. Cole, D. Wang, A. Warren and Y.I. Leiderman, “Feature Tracking and Segmentation in Real Time via Deep Learning in Vitreoretinal Surgery: A Platform for Artificial Intelligence-Mediated Surgical Guidance”, *Ophthalmology Retina*, Vol. 7, No. 3, pp. 236-242, 2023.