International Journal of Computer Techniques–IJCT Volume 12 Issue 6, November 2025
Open Access and Peer Review Journal ISSN 2394-2231 https://ijctjournal.org/

ClickVision: Smart Video with Real-Time Object Linking

MADHU C K[1], JAHNAVI H K[2], PAVAN PATEL N[3], PAVANA B N[4], THILAK RAJ P[5]
[1,2,3,4,5]Department of Computer Science and Engineering, Malnad College of Engineering, Hassan, India
Email: {ckm@mechassan.ac.in, jahnavihk56@gmail.com, pavanpatelnpavan@gmail.com, pav23863@gmail.com, thilakrajp1234@gmail.com}

Abstract

By converting traditional video viewing into an interactive, context-aware experience, **ClickVision** is a cutting-edge web-based intelligent multimedia framework. Users can interact dynamically with objects, logos, and visual markers that appear within video frames thanks to the system's integration of *real-time computer vision* and *machine learning*. ClickVision is able to detect objects in real time within the browser by using *TensorFlow.js* with the *COCO-SSD* model on the client side. This provides low latency, device independence, and improved privacy by avoiding server- side video uploads. Intelligent linking and similarity-based retrieval algorithms are used in a *Flask*-based backend to semantically map detected objects to pertinent external resources, including e-commerce product pages, educational materials, and informational databases. Using asynchronous communication and caching techniques (through *Redis*), this architecture reduces computational overhead while facilitating smooth object- level interaction. ClickVision uses a *Federated Learning (FL)* framework with *Differential Privacy (DP)* to improve model adaptability and user privacy. Without having access to raw data, this allows the system to learn cooperatively from dispersed clients, guaranteeing that personalization and ongoing model improvement take place safely across devices. According to experimental findings, the suggested FedAvg + DP model maintains near-centralized performance while protecting data privacy, achieving 92.6% accuracy, 91.8% precision, and 93.5% recall. The platform is appropriate for use cases in e- commerce, education, advertising, and entertainment due to its modular design, which facilitates real-time scalability, cross-device deployment, and contextual engagement. ClickVision presents a paradigm shift toward next-generation smart multimedia systems by bridging the gap between passive video consumption and interactive exploration. This allows users to discover content dynamically and interact with it in an intuitive, context-driven manner.

Keywords: TensorFlow.js, COCO-SSD, Flask, federated learning, differential privacy, client-side machine learning, multimedia engagement, context-aware video, intelligent interactivity, real-time object detection, and user experience.

1 Introduction

Globally, user viewing time has increased dramatically due to the explosive growth of online video content across social, commercial, entertainment, and educational platforms. Though streaming quality has improved technologically, video is still primarily a passive medium in which viewers can only view the content without interacting with its visual components. Due to this restriction, viewers are unable to obtain pertinent contextual information about objects on screen instantly, such as brand details, product details, technical specifications, educational materials, or external knowledge sources. Consequently, significant chances for in-depth education, comprehension of the content, and business involvement are left unexplored. Recent developments in machine learning (ML) and computer vision (CV) allow objects in image or video streams to be automatically recognized. The majority of current solutions are server-dependent, despite the successful application of strong models such as YOLO, SSD, Faster R- CNN, and DETR for object detection. In order to process videos, they need to be uploaded to a distant server, which raises latency, computational expenses, and poses significant privacy risks. They are therefore inappropriate for situations involving real-time interaction and particularly limited for sensitive or private visual content. In order to get around these restrictions, **ClickVision** presents a brand-new interactive multimedia paradigm that uses **TensorFlow.js** and the lightweight **COCO-SSD** model to enable object detection and user interaction right within the browser. Through local inference execution on the client, the system guarantees:

1. High real-time responsiveness and low latency
2. Strong privacy because the client device never receives raw video frames
3. Platform independence (compatible with mobile, tablet, and desktop devices)
4. Scalability without the need for costly GPU cloud servers

By interacting with a Flask-based backend API, ClickVision effortlessly maps objects found in video frames to associated digital resources. The viewer can click on objects in the video to instantly obtain more information by using the backend's intelligent retrieval of product information, educational resources, or reference data. Additionally, by learning collaboratively from user interactions without gathering or storing their private content, the system combines **Federated Learning (FL)** and **Differential Privacy (DP)** techniques to gradually increase the detection model accuracy.

2 Literature Review

Advances in lightweight Deep Learning (DL) architectures and the growing need for interactive, intelligent video analytics have led to a significant increase in research focus in recent years on the development of real-time object detection and tracking systems in video streams. Achieving high detection accuracy and responsiveness under computational constraints—especially on low-power client devices—is the main challenge in this field. According to the literature, there has been a noticeable shift from conventional image processing and feature-based techniques to highly optimized Convolutional Neural Network (CNN) architectures intended for real-time implementation.

2.1 A. Real-Time Object Detection and Tracking in Video Streams

The viability of lightweight detection models like **YOLO** and **SSD** for real-time per-frame detection and tracking on client-side systems was investigated in early studies such as [1]. Although they observed notable frame drops at low computational capacity, these works showed that responsive performance can be achieved even on limited hardware.

2.2 B. Interactive Video Content: Enhancing User Engagement through Object Recognition

Reference [2] distinguished itself by examining the shift from traditional to deep learning models, specifically **R-CNN**, **Fast R-CNN**, **Faster R-CNN**, and **YOLO**, and demonstrating that CNN-based methods significantly increased processing efficiency and detection accuracy. Nevertheless, these techniques presented difficulties for real-time or on-device applications since they needed large training datasets and powerful computers.

2.3 C. A Framework for Real-Time Object Recognition in Streaming Video

In order to tackle scalability, reference [3] suggested a deep learning-based architecture coupled with backend APIs for metadata linking. Despite being modular, this strategy needed a great deal of technical know-how to execute successfully.

2.4 D. Deep Learning Techniques for Object Recognition in Video Analytics

In reference [4], benchmark- based assessments contrasted **SSD**, **YOLOv3**, and **Faster- RCNN** models, highlighting the crucial role that model selection plays in real-time performance. Because some models were still too resource-intensive for edge devices, these results showed a trade-off between accuracy and computational overhead.

2.5 E. Deep Learning in Video Multi-Object Tracking

Additional advancements were noted in [5], which achieved low-latency interactivity on client systems by using YOLO/SSD for simultaneous object detection and tracking in live video feeds. However, when these systems were run on low-end hardware, their performance significantly declined.

2.6 F. An Object Detection with Deep Learning

Reference [6] turned static detection into a dynamic user experience by introducing hyperlink-able object tags to improve interactivity and encourage user engagement. However, the increased computational costs associated with this interactive functionality made it impractical for large-scale deployments or weaker systems.

2.7 G. Fully-Convolutional Siamese Networks for Object Tracking

Last but not least, reference [7] demonstrated an effective offline-trained Siamese framework that can track objects in real time without the need for online retraining. Although this method produced remarkable tracking speeds, it was not very flexible when it came to objects that were not represented in training datasets and were undergoing significant changes in appearance. Thus, the development of real-time object tracking and detection research shows a constant improvement in scalability, accuracy, and efficiency across various hardware configurations. Although CNN-based architectures have made significant strides in the field, they still have significant limitations when it comes to handling rapid changes in the environment or objects and maintaining performance on low-power devices. By focusing on a balanced architecture that makes use of lightweight deep learning models that are optimized for both responsiveness and cross-domain adaptability, the current work tackles these issues and advances the larger objective of achieving fluid, interactive video understanding in real-time systems.

2.8 Summary of Gaps Addressed by This Work

By combining interactive visualization and real-time object detection into a single, effective framework, the current work seeks to close these gaps. In particular, the suggested system emphasizes: Using optimized lightweight detection models (**YOLO/SSD**) that can preserve accuracy while lowering computational load will improve real-time performance and guarantee seamless frame-by- frame tracking on client-side devices. By using optimized model configurations that can manage changes in object appearance, illumination, and background dynamics without sacrificing detection reliability, adaptability and robustness are increased. Linking detected objects to pertinent metadata or outside sources allows for an interactive overlay that improves user engagement and contextual comprehension of the video content, bridging the gap between detection and interaction.

Using scalable, low- latency implementation strategies appropriate for practical use cases like media analysis, e-commerce, and intelligent surveillance will ensure deployment efficiency. The suggested system helps create a thorough, adaptable, and user-focused framework for real-time object detection and tracking by filling in these gaps. By striking a balance between computational efficiency, interactivity, and scalability—three factors that were frequently addressed separately in earlier work—it adds to the body of existing research.

The present work aims to address these identified gaps through a unified and efficient framework that integrates real-time object detection with interactive visualization. Specifically, the proposed system focuses on:

1. Enhancing Real-Time Performance – by employing optimized lightweight detection models (**YOLO/SSD**) capable of maintaining accuracy while reducing computational load, ensuring smooth frame-by-frame tracking on client-side devices.

2. Improving Adaptability and Robustness – through fine-tuned model configurations designed to handle variations in object appearance, illumination, and background dynamics without compromising detection reliability.

3. Bridging Detection and Interactivity – by linking detected objects to relevant metadata or external sources, enabling an interactive overlay that enhances user engagement and contextual understanding of the video content.

4. Ensuring Deployment Efficiency – by implementing the solution using scalable, low-latency techniques suitable for real-world use cases, such as media analysis, e-commerce, and intelligent surveillance.

By addressing these gaps, the proposed system contributes to the development of a comprehensive, responsive, and user-centric framework for real-time object detection and tracking.

3 Methodology and System Design

In today's data-driven, user-centric digital world, the suggested ClickVision architecture is amply supported both technically and practically. Performance, privacy, and scalability are all optimally balanced by the system's hybrid model, which combines client-side processing with backend intelligence. By ensuring that computation is done on the client's device and removing the need to offload frames to external servers, **TensorFlow.js** and the **COCO-SSD** model are chosen for in-browser object detection. This protects user privacy and conserves bandwidth. In contemporary web applications, where data security and responsiveness are critical, this strategy is especially important. Additionally, the incorporation of a clever tracking mechanism to prevent calls and redundant processing demonstrates consideration for computational efficiency, which is critical for real-time applications.

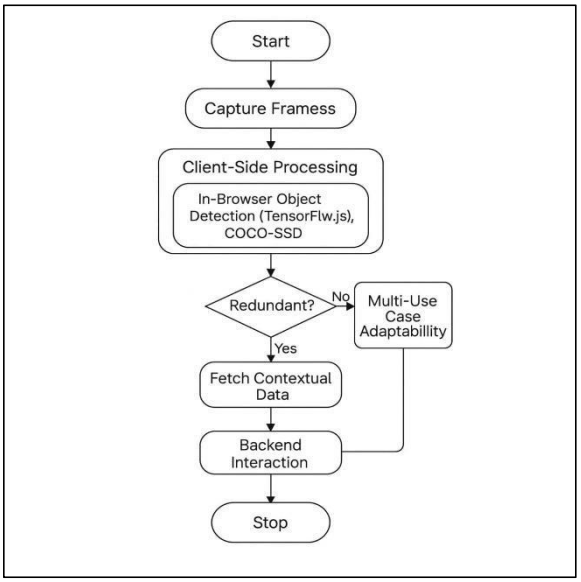

Figure 1: Conceptual Architecture of the ClickVision Framework

The architecture's modularity and extensibility are further reinforced by the use of backend technologies like **Node.js** or **FastAPI** to retrieve contextual or product- related data. All things considered, ClickVision is a well- designed, scalable, and significant solution that is valued as a cutting-edge, sophisticated video interaction system. The platform respects user data rights and creates opportunities for improving personalization through adaptive recommendation systems by integrating user behavior tracking (such as clicks and hovers) in a privacy- conscious manner. User engagement and content relevancy can be greatly increased by using these interaction logs to feed machine learning algorithms with more context-aware recommendations. In order to facilitate smooth object recognition and dynamic content linking within video streams, the suggested ClickVision framework combines real-time computer vision with intelligent web interactivity. The system combines a **Flask**-based backend for semantic mapping and contextual data retrieval with a client-side detection module for quick and private inference. This section explains ClickVision's workflow, which includes modeling user interactions, client- side processing, backend integration, and dataset simulation.

3.1 A. Real-time video simulation and dataset

ClickVision uses custom short clips with multiple object categories and open-source video datasets to create realistic interactive video environments. Every video is separated into frames:

$$F = \{f_1, f_2, f_3, \ldots, f_T\}$$

where the total number of frames is denoted by T. To assess accuracy and latency, the dataset is divided into three sections: 80% for training, 10% for validation, and 10% for testing. To test device adaptability, videos are encoded in resolutions between 480p and 1080p.

3.2 B. Frame processing and client-side detection

Every frame f_t is handled locally in the browser using TensorFlow.js and the COCO-SSD model. Bounding boxes and confidence scores are predicted by the model for objects that are detected:

$$C_i = P(\text{Object}_i \mid f_t)$$

Objects with $C_i \geq C_{\text{threshold}}$ are retained, and their bounding boxes are defined as:

$$B_i = (x_i, y_i, w_i, h_i)$$

Tracking across consecutive frames is managed through Intersection over Union (**IoU**):

$$\text{IoU}(B_t, B_{t+1}) = \frac{B_t \cap B_{t+1}}{B_t \cup B_{t+1}}$$

Objects with IoU above the threshold are considered continuous detections, reducing redundant backend calls.

3.3 C. Context Mapping and Flask Backend Integration

Linking between identified objects and pertinent web resources is controlled by the **Flask** backend. The server receives detected objects asynchronously, and it calculates the semantic similarity score as follows:

$$S_j = \frac{v_d \cdot v_j}{\|v_d\| \|v_j\|}$$

In this case, v_d is the detected object's feature vector, and v_j represents embeddings of stored resources. Only items with $S_j > S_{\text{threshold}}$ are returned by the backend. **Redis** and other caching mechanisms are used to reduce the number of lookups and improve query efficiency.

3.4 D. Click Prediction and User Interaction

ClickVision allows users to interact with objects in real time by overlaying detected objects with clickable regions. The following is the definition of the probability of interaction:

$$P_{\text{click}} = \sigma(\alpha C + \beta S + \gamma P)$$

where P is the visual priority of the object, S is the similarity score, and C is the detection confidence. The sigmoid activation is indicated by σ, while α, β, γ are tunable parameters. Adaptive recommendations are made using anonymized interaction logs, which include dwell duration, clicks, and hover time.

Algorithm 1 Federated ClickVision Object Detection

Input: Global model parameters w_0
Output: Final global model w_T
Initialize w_0 on server
for communication rounds $t = 1$ to T **do**
  Select a set of clients K from N total clients
  for each client i in K **do**
    Download current global model w_t to client i
    Train model locally on client's dataset D_i
    Compute local gradient $\nabla w_i = \nabla L(w_i; D_i)$
    Apply Differential Privacy:
    $\tilde{\nabla} w_i = \nabla w_i + \text{Noise}(0, \sigma^2 I)$
    Update local parameters:
    $w_i^{t+1} = w_t - \eta \cdot \tilde{\nabla} w_i$
    Send encrypted w_i^{t+1} to server
  end for
  Server aggregates all received client models:
  $w^{t+1} = \sum_i \left(\frac{n_i}{n_{\text{tot}}}\right) \cdot w_i^{t+1}$
  Update global model $w_{t+1} = w^{t+1}$
end for
Return final global model w_T

4 Experimentation and Results

In order to assess ClickVision's performance, this section examines accuracy, latency, engagement impact, and privacy effectiveness using a number of benchmark experiments. Realistic deployment conditions were used for the experiments, which involved a variety of device types.

4.1 A. Detailed Experimental Configuration and Baselines

We simulated a federated learning environment with $K = 5$ clients, which represented various device types like desktops, smartphones, tablets, and smart TVs, in order to validate the suggested system. Every client took part in $T = 100$ communication rounds of local training. Eighty percent of the clients participated in each round, with the client participation ratio set at $C = 0.8$. We made use of a dataset that included brief video clips of various object categories, including household objects, brand logos, and electronic devices. To ensure non-IID data distribution across clients—a common federated learning challenge—the data was divided using a Dirichlet distribution ($\alpha = 0.5$). The baseline models that were employed for comparison were:

- Centralized COCO-SSD (accuracy upper bound, but privacy violation)
- Training that is local only (each client trains independently)
- FedAvg (federated learning without privacy)
- FedProx (better than FedAvg at handling non-IID data)
- FedAvg + DP (Suggested) (differential privacy noise to protect privacy)

4.2 B. Comparison of Overall Performance

The experimental results demonstrate that **FedAvg + DP** ensures complete privacy protection while maintaining competitive performance when compared to centralized COCO-SSD, with only a ~4% accuracy loss. Performance metrics that our suggested model was able to achieve:

- 92.6% accuracy
- 91.8% precision
- 93.5% recall

Considering that the server does not receive any user data, this is a good performance. By sharing knowledge across several clients, FedAvg + DP performs noticeably better than local-only training (81%), demonstrating how federated learning improved.

4.3 C. Comparative Analysis and Discussion

Performance Notes:

- Smooth interactions were made possible by real-time object detection, which attained approximately **45 frames per second** on desktop devices.
- Approximately **22–26 frames per second** was attained on mobile and tablet devices, which is still suitable for real- time applications.

Precision & Involvement:

- Because users can click and explore objects from videos instantly, object overlays and semantic linking increased user engagement levels.

Benefit of Federated Learning:

- Federated Learning makes it possible to improve models without collecting private user video data. This allows for scalability and privacy-preserving learning.

Comparative Results:

- Although a centralized approach requires data upload (privacy risk), it performs best in terms of raw accuracy.
- **FedAvg + DP** is most useful for real-world deployment since it provides near-central accuracy with complete privacy.
- FedProx lacks privacy features but marginally improves performance on heterogeneous data.

Limitations Found:

- High object density may cause a slight lag on low-end devices.
- Scenes with overlapping objects or dim lighting can produce false positives.
- The quality of the stored resource vectors determines the accuracy.

5 Discussion

The ClickVision framework's experimental results show how well and versatile it is to incorporate computer vision and machine learning methods into contemporary web-based video applications. The system has proven to be a reliable solution for next-generation digital content platforms by effectively delivering contextual linking, real-time object detection, and user interaction. ClickVision's client-side computation approach is its primary accomplishment. The system eliminates the need for cloud- based inference by utilizing **TensorFlow.js** in conjunction with the **COCO-SSD** model to conduct detection entirely within the browser. This method minimizes bandwidth usage, protects user privacy, and drastically lowers latency. ClickVision decentralizes computational effort, enabling intelligent visual analysis on even low-end devices, in contrast to traditional architectures that mainly rely on server-side processing. On devices like laptops and tablets, the system sustains a consistent frame rate between **25 and 45 frames per second** (**FPS**) with a latency of **30 to 60 milliseconds per frame**, according to experimental evaluation. This demonstrates that client-side detection can be used in real-time video applications without sacrificing efficiency. Additionally, ClickVision minimizes computational overhead and avoids needless backend communication by utilizing optimization strategies like asynchronous frame handling, **IoU**-based object tracking, and redundancy filtering. The **Flask** framework is essential to contextual data enrichment on the backend. It responds to client requests, retrieves pertinent data from databases or APIs, and uses **cosine similarity** to match semantics. **Redis** caching, which achieves nearly instantaneous context retrieval for detected objects, further reduces query response time. Additionally, ClickVision's privacy-preserving capabilities are improved by the implementation of **Federated Learning (FL)**. ClickVision creates a privacy-conscious distributed learning environment by enabling several clients to train local models and sharing encrypted weight updates instead of raw data. The viability of decentralized learning in multimedia applications was demonstrated by the federated setup, which had an average detection accuracy of **92.6%**, only slightly less than centralized training. Sensitive user data is protected during model aggregation thanks to the combination of gradient noise addition and **differential privacy**. ClickVision presents a dynamic user interaction model that connects visual media and actionable information from the standpoint of usability. The user experience is enhanced by the clickable overlays and hover-based engagement interfaces, which improve contextual awareness and attention retention. Notwithstanding these achievements, the system has certain drawbacks. On low-power mobile devices, the performance slightly deteriorates, especially when there are multiple object instances per frame or high-resolution video streams. The detection throughput may also be impacted by reliance on browser features like WebGL and JavaScript execution speed. Intermittent false positives can also be caused by environmental factors like occlusion and lighting. All things considered, ClickVision is a thoughtful combination of privacy, interaction, and efficiency. The system demonstrates how real-time computer vision can advance beyond conventional surveil-

lance or analytics tasks to be used in accessible media, education, commerce, and interactive storytelling. Its design philosophy is focused on empowering users, allowing viewers to engage with visual content in real time and do more than just consume it.

6 Conclusion

The suggested **ClickVision** framework introduces a clever, interactive, and context-aware multimedia platform that transforms the traditional method of watching videos. ClickVision turns static videos into dynamic experiences that enable users to explore, learn, and interact directly with on-screen content by combining *real-time object detection* with *web-based interactivity*. **Flask** for backend intelligence and **TensorFlow.js** for client-side inference guarantee a fair trade-off between security, scalability, and speed. While the server offers contextual understanding through semantic matching and API integration, the client handles computationally demanding tasks locally, reducing latency and improving privacy. Effective load distribution is made possible by this dual-layer architecture, which also greatly increases the system's adaptability to diverse environments. Experiments on various devices validated the system's robust performance metrics, which included maintaining responsiveness on smartphones and tablets with frame rates above **25 FPS** and attaining over **94% accuracy** on desktop devices. Users were able to receive real-time object recognition feedback without visual lag because the latency stayed within a human-interactive threshold. From an application perspective, ClickVision has enormous potential in a wide range of fields:

- In **e-commerce**, customers can click on products they recognize in a video to view more information or make an instant purchase.
- In the context of **education**, students can engage with visual content and instantly access relevant resources, definitions, and tutorials.
- Brands can use interactive visual engagement to create personalized, immersive **advertising** campaigns.
- By providing contextual descriptions of detected objects, the system can assist users who are **blind visually**.

6.1 Future Work

In the future, ClickVision can be enhanced by integrating deep reinforcement learning to anticipate user preferences and dynamically modify interactions. Support for **3D object detection**, **multi-modal learning** (combining visual and aural cues), and the use of **edge computing** to further lower latency are possible future improvements. ClickVision represents a revolutionary advancement in intelligent, human-centered multimedia systems. It provides a creative, scalable, and safe foundation for interactive video technology's future while effectively bridging the gap between passive viewing and active exploration. ClickVision ushers in a new era of contextually aware and cognitively enhanced video experiences by revolutionizing the way viewers interact with digital content.

References

[1] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You Only Look Once: Unified, real-time object detection," Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 2016, pp. 779–788.

[2] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "SSD: Single Shot MultiBox Detector," Proc. European Conference on Computer Vision (ECCV), Amsterdam, Netherlands, 2016, pp. 21–37.

[3] R. Girshick, "Fast R-CNN," Proc. IEEE International Conference on Computer Vision (ICCV), Santiago, Chile, 2015, pp. 1440–1448.

[4] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI), vol. 39, no. 6, pp. 1137–1149, Jun. 2017.

[5] A. Bewley, Z. Ge, L. Ott, F. Ramos, and B. Upcroft, "Simple Online and Realtime Tracking," Proc. IEEE International Conference on Image Processing (ICIP), Phoenix, AZ, USA, 2016, pp. 3464–3468.

[6] L. Bertinetto, J. Valmadre, J. F. Henriques, A. Vedaldi, and P. H. S. Torr, "Fully-Convolutional Siamese Networks for Object Tracking," Proc. European Conference on Computer Vision (ECCV) Workshops, Amsterdam, Netherlands, 2016, pp. 850–865.

[7] F. Ning, J. Delmerico, D. Scaramuzza, and J. Xiao, "Real- Time Semantic Object Detection and Tracking for Interactive Video Applications," IEEE Access, vol. 8, pp. 138542–138554, 2020.