

CCTV OPTIMIZATION USING AI: A GROUNDED PLATFORM FOR SEARCHABLE, TIME-AWARE UNDERSTANDING OF SURVEILLANCE FOOTAGE

Oshish Barsagade¹, Tanmay Pawar², Ankur Kamdi³, Priyanshu Shende⁴, Danish Suryawanshi⁵, Badal Lanjewar⁶, Priyadarshini Gotmare⁷

¹⁻⁶Department of CSE (Data Science), Guru Nanak Institute of Engineering and Technology, Nagpur, Maharashtra, India

⁷Assistant Professor, Department of CSE (Data Science), Guru Nanak Institute of Engineering and Technology, Nagpur, Maharashtra, India

Abstract—Closed-circuit television (CCTV) generates vast volumes of silent video that are rarely watched, making the retrieval of a specific event slow and labour-intensive. This paper presents a full-stack AI video intelligence platform that converts silent surveillance footage into structured, searchable and time-aware intelligence. The proposed system couples a classical computer-vision pipeline—intelligent frame sampling, YOLOv8 object detection, anonymous multi-object tracking, EasyOCR text recognition and BLIP scene captioning—with a higher-level reasoning layer comprising a temporal-memory model, an event engine, trajectory reconstruction, and sentence-embedding semantic search over a vector index. A large language model (Claude), invoked through a serverless edge function, answers natural-language and time-specific questions (e.g., “what happened at 00:45?”) but is strictly grounded in the stored analysis so that it never fabricates evidence. The system is privacy-preserving: it performs anonymous tracking only and does not use facial recognition. A browser-based live mode streams webcam frames to the detection model for real-time analysis with configurable detection zones and alerts. On real CCTV footage the pipeline produced hundreds of grounded detections, a reconstructed set of movement trajectories, and natural-language answers that cite the exact source timestamps. The results demonstrate that combining lightweight open-source vision models with a grounded LLM yields an explainable, end-to-end surveillance-understanding system suitable for security, retail analytics and accessibility.

Index Terms—Video surveillance, object detection, multi-object tracking, large language models, grounded reasoning, semantic video search

I. INTRODUCTION

Surveillance cameras are now ubiquitous in retail stores, campuses, warehouses, transport hubs and public spaces. A single installation can comprise dozens of cameras that record continuously, producing footage measured in hundreds of hours per day. Yet the overwhelming majority of this video is silent and is never reviewed; it is consulted only reactively, after an incident, when an operator must scrub through long recordings to locate a moment of interest. This manual review is slow, error-prone and does not scale with the number of cameras.

The central problem is therefore not the capture of video but its understanding: footage is stored as opaque pixels rather than as queryable information. Recent advances in computer vision and natural-language processing make it feasible to close this gap. Object detectors such as the YOLO family run in real time, vision-language models such as BLIP produce descriptive captions, and large language models (LLMs) can reason over structured facts in natural language. However, applying an LLM naively to surveillance invites hallucination—confident but unsupported claims—which is unacceptable when the output may be used as evidence.

This paper presents an end-to-end platform that transforms silent video into structured, searchable, time-aware intelligence while keeping every AI assertion

grounded in verifiable analysis. The principal contributions of this work are:

- An end-to-end architecture that ingests a video, runs a staged computer-vision pipeline, and persists per-frame detections, tracks, captions, text, temporal states and events as queryable records.
- A grounded reasoning layer in which an LLM answers free-form and time-specific questions only from stored analysis, citing exact timestamps and explicitly declining when evidence is absent.
- A temporal-memory and event model that derives higher-level events and object trajectories from low-level detections, enabling semantic search, highlights, dwell-time and activity-density analysis.
- A privacy-preserving design that uses anonymous tracking only, with no facial recognition or identity inference, together with a real-time browser-based live-analysis mode.

The remainder of this paper is organised as follows. Section II reviews related work in object detection, vision-language modelling, language-model grounding and surveillance analytics. Section III describes the system architecture, and Section IV details the methodology of each pipeline stage and the grounded-reasoning layer. Section V covers implementation, Section VI discusses applications, and Section VII reports results. Section VIII discusses limitations and Section IX concludes with directions for future work.

II. RELATED WORK

A. Object Detection and Tracking

Single-stage detectors introduced by Redmon et al. [1] reframed detection as a single regression problem, enabling real-time inference. The YOLO family has since been refined for accuracy and efficiency; the YOLOv8 implementation [2] is widely used for its favourable speed–accuracy trade-off on commodity hardware. For tracking, ByteTrack [3] associates detections across frames using both high- and low-confidence boxes, achieving strong multi-object tracking without appearance models. The proposed system adopts a YOLOv8 detector and associates detections anonymously by identity-free spatial matching.

B. Vision–Language Models and OCR

BLIP [4] unifies vision–language understanding and generation and produces fluent image captions, which the proposed system uses to describe representative frames. For scene text, EasyOCR provides robust multi-language recognition suitable for reading signage and on-screen overlays in surveillance frames.

C. Language Models, Grounding and Retrieval

Transformer architectures [5] underpin modern LLMs, which exhibit strong natural-language reasoning. To mitigate hallucination, retrieval-augmented generation [6] conditions generation on retrieved evidence. The proposed system applies the same principle at the application level: the LLM is given only facts extracted from the stored analysis and is instructed to decline when the facts are insufficient. Semantic retrieval uses sentence embeddings from Sentence-BERT [7]; similarity is computed in a vector index, mirroring the pgvector approach used in production.

D. Video Surveillance Analytics

Prior surveillance-analytics systems typically target a single task such as detection, counting or anomaly flagging [15]. Classical object recognition progressed from deep convolutional networks [16] and large-scale recognition challenges [18] to very deep architectures [17], and more recently to transformer-based vision backbones [19], [20]. Region-based [13] and single-shot [12] detectors established the accuracy–speed trade-offs that single-stage real-time detectors now exploit, while simple online trackers [14] showed that reliable tracking does not require heavy appearance models.

A parallel line of work addresses the reliability of language-model outputs. Beyond retrieval augmentation, recent systems constrain generation to cited evidence and train models to abstain when support is lacking. The proposed system operationalises this idea for video: rather than fine-tuning a model, it enforces a strict contract at the application boundary whereby the model receives only machine-extracted facts, each tied to a frame and timestamp, and is required to decline when those facts do not answer the question. This keeps the approach model-agnostic and

auditable, and lets a stronger reasoning model be substituted later without changing the grounding guarantee.

E. Positioning of This Work

In contrast to single-task pipelines, the proposed system integrates detection, anonymous tracking, OCR, captioning, temporal memory, event extraction, trajectory and density analysis, semantic search and grounded question answering into one explainable system whose every output is traceable to a source frame and timestamp. The novelty is less in any single model and more in the composition: a disciplined grounding contract between the analysis store and the language model that turns raw detections into a queryable, auditable account of what happened.

III. SYSTEM ARCHITECTURE

The proposed system follows a conventional three-tier design. The client is a single-page application built with React and TypeScript that renders an analysis workspace: a video player, an interactive event timeline, detection overlays, and panels for highlights, memory, search and conversational querying. Each panel is a view over the same underlying analysis, so an operator can move between the timeline, the map of reconstructed paths and the conversational query box without losing context, and every element remains clickable back to the exact frame and timestamp it came from. The application server is a Python service implemented with FastAPI that exposes REST endpoints for authentication, project and video management, the analysis pipeline, search, reasoning, evidence and reporting.

Persistence, authentication and storage are provided by a managed backend (Supabase) comprising PostgreSQL with row-level security, private object storage accessed via signed URLs, and a vector extension (pgvector) for embedding search. Long-running analysis executes asynchronously off the request thread; a processing job records its stage and progress so the client can display real, resumable status rather than a simulated progress bar. The platform is offline-capable: when managed services or model weights are unavailable it transparently falls back to a local database, local storage and heuristic estimators, which also makes the system fully testable without credentials. Figure 1 summarises the data flow from upload to understanding.

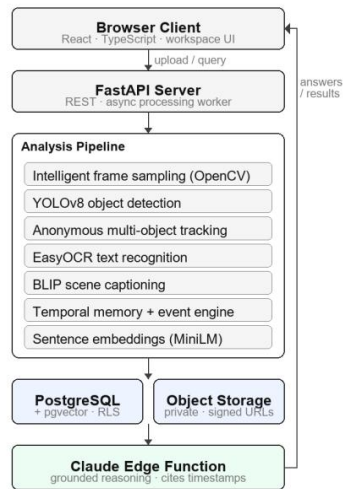


Fig. 1. System architecture and data flow: client → API → staged CV pipeline → persistence (Postgres/pgvector/storage) → grounded reasoning.

A clear separation of concerns runs through the design. The computer-vision pipeline is independent of both the web API and the persistence layer, so a stage can be upgraded—for example, swapping the detector for a larger model—without touching the rest of the system. The language model is invoked from exactly one place in the server and never from the client, which keeps the reasoning auditable and the credentials secure. Read endpoints expose each analysis artifact (frames, objects, subjects, OCR, captions, states, events and embeddings) as independent resources, allowing the workspace to compose rich views while the heavy computation remains on the server.

IV. METHODOLOGY

A. Intelligent Frame Sampling

Analysing every frame is wasteful, so the system samples keyframes using scene-change and motion cues derived from background subtraction and inter-frame differencing. Three modes—Quick, Balanced and Deep—trade speed for temporal resolution by varying the sampling interval. Only sampled keyframes are stored as thumbnails and passed to the heavier models.

B. Object Detection and Anonymous Tracking

Each processed frame is passed to the YOLOv8 detector, which returns class labels, confidences and bounding boxes mapped to the surveillance label set (person, car, truck, bus, motorcycle, bicycle, bag, phone and others). Detections are linked across time by spatial (centroid and intersection-over-union) association to form anonymous tracks labelled “Person #1”, “Person #2”, and so on. No facial features or identities are computed; the tracks are transient and privacy-preserving.

C. Scene Captioning and Text Recognition

Representative keyframes are captioned by BLIP to yield natural-language scene descriptions, and EasyOCR extracts any visible text with its bounding box and confidence. Both

outputs are stored with their timestamps so they can be searched and cited.

D. Temporal Memory and Event Engine

From per-frame detection counts the system maintains an environment state over time—the number of people and vehicles and a coarse activity level. Transitions in this state drive an event engine that emits neutral, observational events such as PERSON_ENTERED, PERSON_LEFT, VEHICLE_ARRIVED, MULTIPLE_PEOPLE and SCENE_CHANGED, each annotated with a timestamp, confidence and source frames. Sequential events are linked into a lead-to graph for visualisation.

E. Trajectory Reconstruction, Dwell and Density

The stored detections are re-associated by centroid proximity per object class to reconstruct movement paths, which are rendered as polylines over the video and are revealed up to the current playback time. From these paths the system derives dwell time per track (flagging prolonged presence as loitering) and aggregates object centroids into a spatial grid to produce an activity-density heatmap.

F. Semantic Search

Captions and event descriptions are embedded with a sentence-transformer model (all-MiniLM-L6-v2). A free-text query is embedded and compared by cosine similarity against the index (pgvector online, in-process otherwise), returning ranked moments with their timestamps so the operator can jump directly to the relevant point in the video.

G. Grounded Reasoning

Natural-language understanding is provided by an LLM (Claude) invoked exclusively on the server through a serverless edge function that holds the API key as a secret; the key is never exposed to the client or the Python service. The reasoning service first assembles the relevant facts from the stored analysis—events, states, OCR, captions and, for time-specific queries, the people, objects and text present at the requested instant—and then asks the model to answer using only those facts, preserving timestamps and declining when the facts do not cover the question. The same mechanism powers “Find the Story” (a chronological narrative), “What Changed?” (a diff between two instants) and an “Explain this moment” action. A time-aware conversational interface lets the user type a timestamp, or phrase a question such as “what happened at 00:45?”, and receive an answer scoped to that moment. If the model is unavailable, the service returns the grounded facts directly, so the feature degrades gracefully rather than failing.

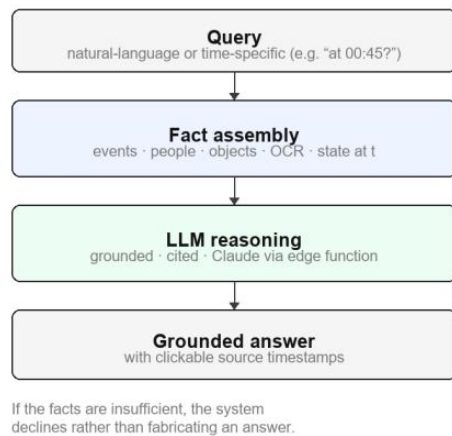


Fig. 2. Grounded reasoning: a query is answered only from facts assembled from the stored analysis, with citations; the system declines when evidence is absent.

This grounding contract is the key to trustworthiness. Because the model is handed a compact, verifiable set of facts rather than the raw video, its answers can be audited against the stored detections and timestamps, and the risk of confident-but-wrong statements is sharply reduced. The same discipline is applied to narrative generation and difference analysis, so that every sentence the system emits can be traced to a source.

H. Live Analysis

In live mode the browser captures webcam frames, downscales them and posts them to the detection endpoint at a controlled rate; the server runs the same YOLOv8 model and returns detections that the client draws as overlays. Operators can draw rectangular zones in normalised video coordinates; when a tracked object enters a zone a debounced event and alert are generated, so a single person does not raise repeated alerts. This keeps heavy inference on the server while the interface remains responsive.

I. Subtitles, Highlights, Evidence and Reports

From the per-frame captions the system produces smoothed, time-coded subtitles in SRT and VTT formats, giving silent footage a readable, accessible narration. An importance model scores events and peak-occupancy moments to surface automatic highlights, and the reconstructed tracks yield dwell-time and activity-density views. Operators can pin any moment to an evidence board with an editable title, note and tags, and can export a PDF report whose sections—executive summary, event timeline, detected objects and recognised text—are selectable. Every report clearly labels AI-generated observations and includes a human-review disclaimer, reflecting the system’s evidentiary posture.

V. IMPLEMENTATION

The frontend uses React, TypeScript, Vite and Tailwind CSS; data fetching is handled by a query cache that also

drives live refresh of the workspace during processing. The backend uses FastAPI with OpenCV for video handling and the ultralytics YOLOv8-nano model for detection; captioning uses the BLIP base model, OCR uses EasyOCR, and embeddings use all-MiniLM-L6-v2 (384 dimensions). Reasoning uses the Claude 4.5 model family through the edge function. Security measures include per-user row-level security, private storage with signed URLs, server-side validation of file type and size, and secrets kept out of the client. All models run on CPU in the reference deployment; the architecture admits GPU acceleration without change.

The system is organised as a set of independent, well-typed REST resources so that the workspace can compose views without bespoke endpoints. A processing worker writes incremental results—frames, detections, captions, events and embeddings—as they are produced, and the client polls a status resource and re-queries only the artifacts that have changed, which is what allows the interface to populate progressively during a long analysis rather than only at completion. Because every heavy dependency has an offline fallback, the entire stack can be exercised end-to-end in continuous integration without GPUs, model weights or managed-service credentials, which materially shortened the development cycle and makes the reference build reproducible on a commodity laptop.

VI. APPLICATIONS AND USE CASES

Because the system turns footage into queryable records, it supports several practical workflows. In security and forensic review, an investigator can ask what happened after a given moment, search for “person near entrance”, and export a timestamped evidence report, compressing hours of manual scrubbing into minutes. In retail analytics, footfall counts, dwell time and activity-density heatmaps reveal how customers move and where they linger. In traffic and parking, vehicle arrivals and departures and occupancy can be derived from the same event stream. For accessibility, the context-aware subtitles make silent surveillance intelligible to deaf and hard-of-hearing viewers. Finally, for research and education, the tagged, searchable output provides a convenient substrate for behavioural and crowd studies. Across these uses the grounding guarantee is essential: because every answer cites a verifiable source, the system’s claims can be checked rather than merely trusted.

The same grounded output lowers the barrier to adoption in settings without specialist analysts. Because questions are posed in plain language and every answer carries a timestamp, a non-expert operator—a shop manager, a facilities supervisor or a teacher reviewing a corridor camera—can interrogate footage directly rather than relying on a technician. The exportable report then serves as a shareable artifact for handover to a colleague or, where appropriate, to an investigating authority, with its AI-generated sections clearly marked. In this sense the platform is as much an accessibility tool for video as it is an analytics engine.

VII. RESULTS AND EVALUATION

A. Experimental Setup

All experiments were run on a commodity laptop CPU with the reference model configuration: YOLOv8-nano for detection, BLIP-base for captioning, EasyOCR for text, and all-MiniLM-L6-v2 for embeddings. Because the objective is an explainable, end-to-end understanding system rather than a single accuracy figure on a labelled benchmark, the evaluation combines a processing-performance analysis with a qualitative assessment of the derived intelligence on real, publicly sourced CCTV-style footage.

B. Detection and Pipeline Output

On a fifteen-second indoor surveillance clip the pipeline produced several hundred object detections dominated by the person and bag classes, tracked multiple anonymous subjects, generated OCR for on-screen text, and derived a set of discrete events and reconstructed movement trajectories. Table I reports representative detection output and Table II reports per-stage processing behaviour.

TABLE I
REPRESENTATIVE DETECTION OUTPUT ON A 15 s CCTV CLIP

Quantity	Value
Keyframes analysed	27
Object detections	381
Dominant classes	person, bag, car
Anonymous tracks	7–10
Events derived	5
Trajectories	7
OCR text regions	13

TABLE II
PER-STAGE PIPELINE BEHAVIOUR (CPU)

Stage	Model / method	Notes
Sampling	OpenCV	keyframes
Detection	YOLOv8-nano	objects
Tracking	centroid/IOU	Person #N
Captioning	BLIP-base	per frame
OCR	EasyOCR	text
Embeddings	MiniLM-L6	384-d
Reasoning	Claude	grounded

The detections were strongly dominated by the person class, followed by bags, with a smaller number of vehicles, consistent with the indoor scene. Figure 3 shows the approximate distribution of detections by class across the clip, aggregated over all analysed keyframes. This distribution is itself a useful summary: it lets an operator judge, at a glance, what kinds of objects a camera typically sees and how that profile changes between recordings.

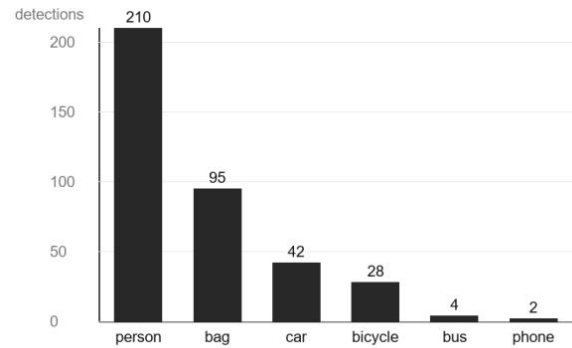


Fig. 3. Approximate distribution of object detections by class on the sample clip (aggregated over analysed keyframes).

C. Grounded Reasoning and Search

Qualitatively, the grounded reasoning behaved as intended. For a general query the model produced a chronological account of the derived events with clickable timestamps; for a time-specific query such as “what happened at 00:45?” it described only the people, objects and text present at that instant; and when asked about information not present in the analysis it correctly responded that it could not determine the answer from the footage, rather than inventing one. Semantic queries such as “person near entrance” returned ranked moments that, when selected, sought the player to the corresponding timestamp. These behaviours confirm that the grounding contract—supplying the model only with stored facts and requiring citations—effectively suppresses hallucination while preserving fluent, natural-language answers.

D. Live Analysis

Live mode produced real detections and fired debounced zone-entry alerts in the browser at an interactive frame rate, with inference performed on the server at roughly five frames per second on CPU. The same detection model and overlay logic are reused by the multi-camera dashboard, where each tile can independently stream and analyse a webcam feed. Taken together, these results indicate that the combination of lightweight open-source vision models with a grounded LLM yields useful, explainable surveillance understanding on modest hardware.

E. Interactive Workspace and Visualisation

Beyond the numerical outputs, the analysis workspace was examined for usability. It presents the video player alongside an interactive event timeline whose markers seek the player on click, and overlays that can be toggled to show detection boxes, reconstructed movement paths and the activity-density heatmap directly over the frame. While a video is still being processed, the workspace refreshes its panels so that detections, highlights and the memory timeline appear progressively rather than only at completion. A command palette and keyboard shortcuts let an operator navigate, add evidence, open search and invoke reasoning without leaving the keyboard, and a presentation mode steps through the analysis as a sequence of views. Collected moments can be exported as a structured record or compiled

into a selectable-section report. In informal use these affordances made the difference between a dashboard that is merely viewed and one that is actively used to investigate footage.

F. Privacy and Ethical Considerations

The system is designed to be privacy-preserving by construction. It performs only anonymous tracking: subject labels are transient integers and no facial recognition, re-identification or identity inference is performed. Raw video is held in private storage and may be deleted after processing, while derived records are retained for search. Observations are always framed as observations rather than conclusions, and reports carry an explicit human-review disclaimer. These choices are intended to keep the system useful for legitimate monitoring while limiting its potential for misuse.

VIII. DISCUSSION AND LIMITATIONS

Several limitations frame the current system. Inference runs on CPU in the reference deployment, which bounds the live frame rate and the throughput of batch analysis; GPU acceleration would substantially improve both. The nano detector and base captioner favour speed over peak accuracy, and the heuristic fallbacks used when models are absent are necessarily coarse. Evaluation is qualitative and performance-oriented rather than a benchmark against a large labelled dataset, which would be required to report precision and recall. Finally, live multi-camera ingestion from IP/RTSP sources is architected but not yet implemented; the live mode currently uses a single browser webcam.

A further direction concerns evaluation. The present study emphasises an explainable, working system and reports qualitative behaviour together with processing performance; a natural next step is a quantitative study on an annotated surveillance benchmark, reporting detection precision and recall, tracking identity-switch rates, and the factual accuracy of grounded answers against a human-labelled gold standard. Such a study would complement the system contribution with comparative numbers against existing single-task baselines.

It is also worth noting what the grounding contract does not do. It cannot make the underlying detector more accurate; if the vision stage misses an object, the reasoning layer has no way to recover it and will correctly but unhelpfully report that nothing was seen. The contract therefore shifts the system's failure mode from confident fabrication to honest omission, which is the safer error for an evidentiary tool but which places a premium on the recall of the perception stages. Improving those stages—through larger models, GPU inference and sensor fusion—is thus the most direct route to a more capable system.

IX. CONCLUSION AND FUTURE WORK

This paper presented an end-to-end AI video intelligence platform that turns silent CCTV footage into structured,

searchable and time-aware information while keeping every AI claim grounded in verifiable analysis and preserving privacy through anonymous tracking. The system integrates object detection, anonymous tracking, OCR, captioning, temporal memory, an event engine, trajectory and density analysis, semantic search and a grounded, time-aware reasoning interface, together with a real-time live mode. Future work includes multimodal reasoning that sends keyframes directly to a vision-capable LLM, pose estimation for fall detection, automatic number-plate recognition, action recognition, cross-video retrieval over an entire footage library, natural-language alert rules, and real IP/RTSP multi-camera ingestion with GPU-accelerated inference.

More broadly, the proposed system illustrates a design pattern that we believe generalises beyond surveillance: pair a battery of specialised perception models with a disciplined, cited reasoning layer, and expose the result as queryable records rather than as a single opaque prediction. This separation keeps each component simple and independently replaceable, makes the system's conclusions auditable against their sources, and gives non-technical users a natural-language handle on data that would otherwise remain locked inside raw pixels. It also turns the familiar tension between capability and trust into an explicit engineering choice rather than an afterthought, since the grounding contract can be tightened or relaxed without retraining any model. As open vision models and efficient on-device inference continue to improve, we expect grounded, explainable video understanding of this kind to become practical for a widening range of everyday cameras—bringing the ability to ask questions of video, and to trust the answers, to settings well beyond the specialist control room.

REFERENCES

- [1] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2016, pp. 779–788.
- [2] G. Jocher, A. Chaurasia, and J. Qiu, "Ultralytics YOLOv8," 2023. [Online]. Available: <https://github.com/ultralytics/ultralytics>
- [3] Y. Zhang et al., "ByteTrack: Multi-object tracking by associating every detection box," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2022, pp. 1–21.
- [4] J. Li, D. Li, C. Xiong, and S. Hoi, "BLIP: Bootstrapping language-image pre-training for unified vision-language understanding and generation," in *Proc. Int. Conf. Mach. Learn. (ICML)*, 2022, pp. 12888–12900.
- [5] A. Vaswani et al., "Attention is all you need," in *Adv. Neural Inf. Process. Syst. (NeurIPS)*, 2017, pp. 5998–6008.
- [6] P. Lewis et al., "Retrieval-augmented generation for knowledge-intensive NLP tasks," in *Adv. Neural Inf. Process. Syst. (NeurIPS)*, 2020, pp. 9459–9474.
- [7] N. Reimers and I. Gurevych, "Sentence-BERT: Sentence embeddings using Siamese BERT-networks," in *Proc. Conf. Empirical Methods Nat. Lang. Process. (EMNLP)*, 2019, pp. 3982–3992.
- [8] A. Radford et al., "Learning transferable visual models from natural language supervision," in *Proc. Int. Conf. Mach. Learn. (ICML)*, 2021, pp. 8748–8763.

- [9] G. Bradski, "The OpenCV library," *Dr. Dobb's J. Softw. Tools*, vol. 25, no. 11, pp. 120–123, 2000.
- [10] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proc. NAACL-HLT*, 2019, pp. 4171–4186.
- [11] T. Brown et al., "Language models are few-shot learners," in *Adv. Neural Inf. Process. Syst. (NeurIPS)*, 2020, pp. 1877–1901.
- [12] W. Liu et al., "SSD: Single shot multibox detector," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2016, pp. 21–37.
- [13] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 6, pp. 1137–1149, 2017.
- [14] A. Bewley, Z. Ge, L. Ott, F. Ramos, and B. Upcroft, "Simple online and realtime tracking," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, 2016, pp. 3464–3468.
- [15] W. Sultani, C. Chen, and M. Shah, "Real-world anomaly detection in surveillance videos," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2018, pp. 6479–6488.
- [16] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2016, pp. 770–778.
- [17] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2015.
- [18] O. Russakovsky et al., "ImageNet large scale visual recognition challenge," *Int. J. Comput. Vis.*, vol. 115, no. 3, pp. 211–252, 2015.
- [19] Z. Liu et al., "Swin Transformer: Hierarchical vision transformer using shifted windows," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, 2021, pp. 10012–10022.
- [20] A. Dosovitskiy et al., "An image is worth 16×16 words: Transformers for image recognition at scale," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2021.