# The Right Pipeline Is All You Need: Intelligent Video Analysis at the Edge

*An AI Experiment by DLYog Lab*

Tarun Chawdhury · Mousumi Chawdhury

*DLYog Lab*

March 2026

Research Preview

🎤 **Audio Overview**

0:00 3:20

## ABSTRACT

**Background.** *Automated video classification for security and safety monitoring is a broadly challenging problem requiring both temporal motion reasoning and fine-grained visual understanding of domain-specific behavioural indicators. Existing deep-learning approaches demand large labelled datasets, significant compute resources, and produce opaque outputs that operators cannot easily inspect or override. Intelligent surveillance systems further demand real-time responsiveness, multi-modal alerting, and increasingly, the ability for non-technical operators to interact with monitoring AI using natural language.*

**Methods.** *We present DLVidYog (DL Video Yog), a prompt-configurable video analysis platform that combines classical image processing — dense optical flow, person localisation via YOLO, and motion heatmap visualisation — with structured subject-matter expert (SME) knowledge encoded as prompts, and PHI-4 [1], a small (14B parameter) multimodal language model with text, vision, and audio capabilities. A web-based platform layer enables non-technical users to create, configure, and switch between fully independent use cases — each with its own per-frame analysis prompt, classification rules, audio analysis prompt, output labels, and motion thresholds — without modifying any code. We demonstrate the pipeline on intruder detection (security surveillance) as the primary use case, with the architecture designed to extend to any observable domain including falls detection, industrial safety monitoring, and behavioural analysis.*

**Results.** *The intruder detection use case, configured entirely through the platform UI using PHI-4-generated prompts, correctly classifies normal pedestrian activity as NORMAL and correctly identifies suspicious behaviours based on operator-defined criteria — demonstrating that the pipeline achieves accurate domain-specific classification without any model retraining. A key architectural bug discovered during platform deployment — hardcoded domain label constants in the classification parser — was identified and corrected, revealing a broader principle about dynamic label resolution in multi-use-case pipelines.*

**Future Vision.** *We outline a proposed extension of DLVidYog to real-time remote monitoring: a mobile client application that connects to a DLVidYog server, receives continuous video feed from IP or smartphone cameras, and supports natural language interaction — enabling security operators to ask questions such as "describe current movement near the back entrance" or issue voice commands via browser-native audio recording transcribed by PHI-4. Automated tiered alerts (push notification, email, emergency services escalation) close the loop between AI classification and human response.*

**Conclusion.** *This work demonstrates that a carefully engineered hybrid pipeline — combining deterministic signal processing, scene detection, SME-encoded domain knowledge, a compact multimodal LLM, and a prompt-configurable platform layer — achieves accurate video classification without large labelled training sets, domain-specific models, or code changes between use cases. The architecture is directly extensible to real-time surveillance, natural language monitoring interaction, and automated multi-channel alerting.*

**Keywords:** video analysis · prompt configuration · intruder detection · security surveillance · optical flow · multimodal LLM · PHI-4 · YOLO · motion-aware frame selection · structured prompting · small language model · domain-agnostic pipeline · audio-visual fusion · real-time remote monitoring · natural language interaction · automated alerts · voice commands · surveillance AI · edge AI · falls detection · industrial safety

---

## 1. INTRODUCTION

Automated video analysis — classifying what is happening in a recording, who or what is involved, and how significant the observed event is — is a broadly applicable problem spanning safety, security, industrial monitoring, and human behaviour research. Across all these domains, a persistent barrier to deployment is the requirement for large, domain-specific labelled training datasets and custom-trained models, which are costly, slow to develop, and cannot easily be adapted by domain experts without machine learning expertise.

The recent emergence of small, capable multimodal language models — models that jointly understand images, audio, and text — opens a new paradigm: *vision-language reasoning* guided by structured domain knowledge encoded as prompts. Rather than training a model for each domain, a single general-purpose multimodal model can be steered toward any classification task by encoding the relevant expert criteria as natural-language instructions. PHI-4 [1], a 14-billion parameter model from Microsoft, demonstrates strong visual and audio understanding with instruction-following reliability that makes it practical for structured output generation.

This paper presents **DLVidYog** (DL Video Yog), a multi-use-case video analysis *platform* with a primary focus on intelligent security monitoring. Any operator can create a new use case — specifying the domain, indicators of interest, classification labels, and analysis rules — entirely through a web UI, with PHI-4 optionally generating an initial prompt configuration from a plain-language description. The same pipeline infrastructure (optical flow, YOLO detection, heatmap overlay, two-stage LLM classification) executes identically for every use case; only the prompt layer changes.

The platform's primary demonstrated use case is **intruder detection** — a security surveillance domain requiring recognition of suspicious behaviour, unusual movement patterns, and access violations. The same architecture extends naturally to other safety and monitoring domains, including falls detection, industrial safety compliance, gait analysis, and neonatal care monitoring — any scenario where observable visual and acoustic indicators can be encoded as structured expert criteria.

Beyond the current platform, this paper outlines a vision for extending DLVidYog to **real-time remote monitoring**: continuous video ingestion from IP cameras or mobile devices, a mobile client application enabling remote observation, and critically, **natural language interaction** — allowing operators to query the running monitor using voice commands processed by PHI-4. This interaction model — asking a live AI monitor *"describe what is happening near the entrance"* or *"has anyone entered the restricted zone in the last minute?"* — represents a novel human-AI interface for surveillance that moves beyond alert dashboards toward conversational situational awareness.

## 2. RELATED WORK

### *2.1 Classical Video-Based Motion Analysis*

Optical flow is a well-established technique for computing per-pixel motion between consecutive video frames. The Farneback dense optical flow algorithm [9] provides robust, physics-grounded motion estimation by modelling local displacement through polynomial signal expansion, making it particularly well-suited to variable frame rates and low-resolution surveillance footage. Motion magnitude profiles — temporal sequences of per-frame flow magnitude — have been shown to characterise how movement events evolve across a video clip, enabling motion-guided frame selection that captures peak-activity frames rather than arbitrary temporal positions. DLVidYog applies this principle directly: the highest-motion frame within each of four temporal windows is selected for downstream LLM analysis, ensuring that transient or brief events are not missed by fixed-interval sampling.

### *2.2 Person Detection and Pose Estimation*

YOLO (You Only Look Once) [2] object detection provides fast, accurate person localisation, enabling region-of-interest cropping that focuses downstream analysis on the subject rather than the environment. In surveillance

contexts, person detection is foundational: Sreenu and Durai [10] review the application of YOLO and related detectors for crowd analysis and abnormal behaviour detection in intelligent video surveillance systems. Body pose estimation systems such as MediaPipe [3] and OpenPose [4] enable landmark-level motion quantification but introduce additional dependencies and failure modes in conditions where occlusion, low resolution, or non-standard postures occur.

### *2.3 Anomaly Detection and Intruder Recognition in Surveillance Video*

Automated detection of abnormal or suspicious behaviour in surveillance video is a well-studied but challenging problem. Sultani, Chen, and Shah [11] proposed a weakly-supervised approach for real-world anomaly detection using only video-level labels, achieving strong results on the UCF-Crime benchmark. Kiran et al. [12] survey deep learning methods for unsupervised and semi-supervised anomaly detection in videos, identifying the scarcity of labelled anomaly data as the defining challenge of the field — a challenge that prompt-driven LLM approaches can circumvent by encoding expert criteria directly as instructions. Liu et al. [13] provide a systematic taxonomy and comparison of deep neural network methods for generalised video anomaly event detection, noting that domain-generalisation remains an open problem for trained models. Liu et al. [29] advanced this work with future-frame prediction for unsupervised video anomaly detection — a training-data-intensive approach that DLVidYog circumvents by replacing trained anomaly classifiers with a prompt-configurable LLM reasoning layer. The Vision Transformer (ViT) [26] architecture, which forms the backbone of PHI-4's visual encoder, has further enabled ViT-based surveillance anomaly models that benefit from large-scale pre-training. DLVidYog combines the representational strength of ViT-based features (via PHI-4) with prompt-encoded expert criteria, enabling operators to define what constitutes "anomalous" behaviour for their specific context without requiring annotated training data.

### *2.4 Multimodal Language Models for Video Understanding*

Large vision-language models such as GPT-4V and Gemini have demonstrated zero-shot visual reasoning capability [5], but their closed-source nature and data transmission requirements preclude use in privacy-sensitive or air-gapped deployment contexts. Small open multimodal models — including LLaVA [6], BioMedCLIP [7], and PHI-4 [1] — have demonstrated competitive performance on visual understanding tasks while enabling fully local deployment. Video-ChatGPT [14] extended the video-language model paradigm to temporal video understanding, enabling natural-language question answering over video content. Video-LLaVA [15] further demonstrated that unified visual representation across image and video modalities substantially improves temporal reasoning. Video-LLaMA [20] introduced joint audio-visual language modelling specifically for video streams, and InternVideo2 [22] established state-of-the-art results on video benchmarks through masked video learning combined with multimodal alignment — both directly relevant to PHI-4's audio-vision integration in DLVidYog. PHI-4's instruction-following reliability, multimodal audio-vision support (built on the Phi-3-Vision architecture [25]), and suitability for structured output formats makes it particularly well-suited for the dual-stage classification prompt design used in DLVidYog. The Phi-3-Vision architecture that underpins PHI-4's multimodal capabilities uses a CLIP-compatible ViT [26] visual encoder with projection layers into the language model, a design validated across multiple vision-language benchmarks.

### *2.5 Prompt Engineering for Structured Classification*

Chain-of-thought prompting [8] and structured output prompting have been shown to substantially improve LLM reliability on multi-step reasoning tasks. Kojima et al. [27] further demonstrated that zero-shot chain-of-thought ("let's think step by step") elicits structured reasoning without requiring labelled examples — foundational to DLVidYog's classification prompt design. Encoding SME knowledge as explicit classification criteria — rather than relying on the model's implicit knowledge — reduces hallucination and improves consistency across domains. Few-shot prompting [16] establishes that providing a single labelled example in the prompt substantially improves classification accuracy, a technique DLVidYog operationalises in its use-case generation prompt, which provides a worked example to guide PHI-4 in generating configuration for new domains. The two-stage prompt design in DLVidYog — separating per-frame indicator extraction from aggregate classification — applies the principle of separating reasoning steps to prevent anchoring bias.

### *2.6 Natural Language Interaction with Surveillance and Monitoring Systems*

The application of natural language interfaces to video surveillance and monitoring systems is an emerging research area. Traditional surveillance systems operate through rule-based alert triggers (motion zones, object crossing lines), requiring operators to pre-specify every condition of interest. The availability of large video-language models capable

of zero-shot temporal reasoning [14, 15, 20, 22] opens the possibility of conversational surveillance interaction: operators querying a running monitor in natural language without pre-programming specific alert conditions. CLIP [23] established the vision-language alignment that underpins text-query-based video retrieval, while Whisper [24] provides the robust speech-to-text backbone for voice command pipelines. Deruyttere et al. [30] explored natural language command interfaces to autonomous vehicles — the closest published analogue to commanding a camera monitoring system via NL — while Gao et al. [17] demonstrated natural language video moment localisation, enabling retrospective queries such as *"when did a person last approach the server room door?"* The novelty of the DLVidYog interaction model (Section 9.3) is the integration of voice-command input (transcribed via PHI-4's native audio capability), LLM-based situational query processing, and continuous per-clip AI analysis into a unified real-time pipeline — eliminating the need for separate specialised models for transcription, visual understanding, and classification.

### *2.7 Edge AI and Remote Monitoring Architecture*

Edge computing architectures — deploying inference close to data sources rather than centralised cloud — are well-established for IoT and surveillance applications [18]. The convergence of capable small language models (PHI-4, LLaVA) with consumer-grade GPU hardware makes local LLM inference practical for facility-scale deployments. Howard et al. [28] established MobileNetV3 as the benchmark for lightweight on-device detection before offloading to a heavier server-side LLM, motivating the two-tier edge/server inference architecture described in Section 9. Mohammed et al. [19] survey mobile video streaming architectures for real-time AI inference, identifying latency and bandwidth as the primary constraints for camera-to-server pipelines. WebRTC and RTSP protocols are the dominant transport layers for low-latency video streaming from IP cameras and mobile devices respectively. DLVidYog's current queue-based architecture (REST upload → PostgreSQL queue → background worker) is a natural foundation for continuous real-time ingestion by introducing a streaming ingest layer upstream of the existing pipeline.

## 3. PLATFORM ARCHITECTURE AND ANALYSIS PIPELINE

DLVidYog operates at two levels: a *platform layer* that manages use case configuration, user sessions, and job routing; and an *analysis pipeline* that executes identically for every use case, parameterised entirely by the active use case's prompt configuration. Figure 1 illustrates the data flow from raw video input to final classification and summary output.

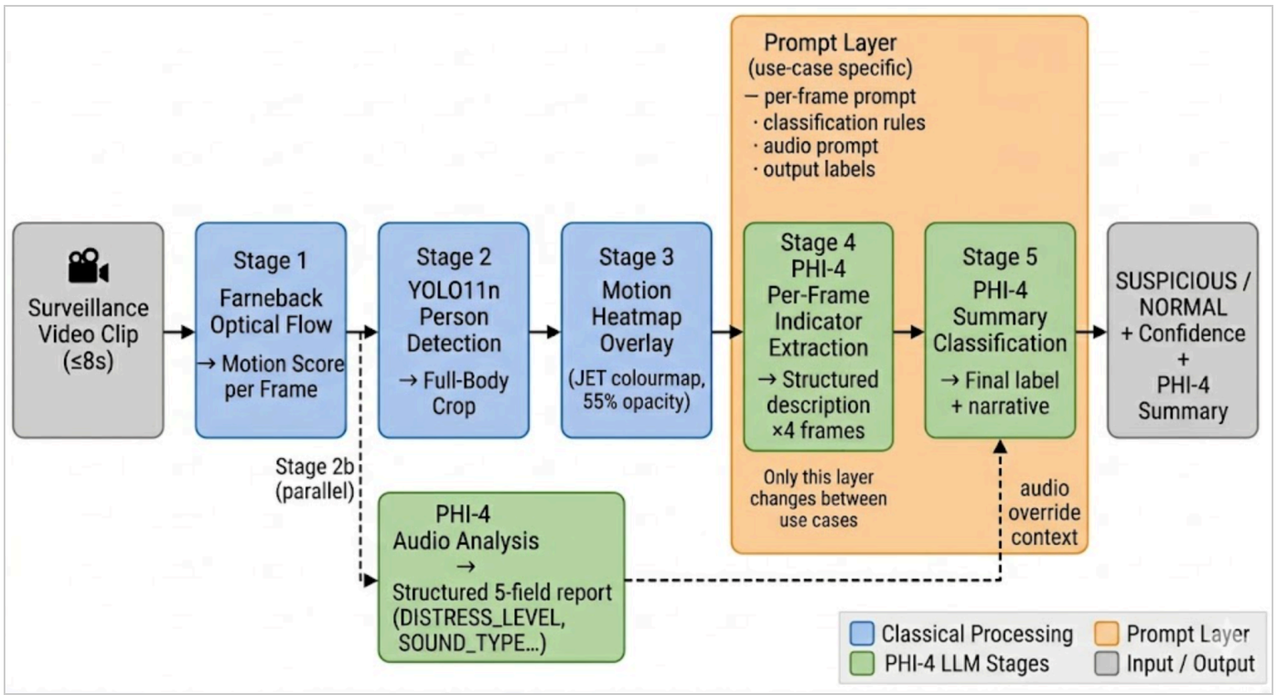


**Figure 1.** The DLVidYog pipeline from raw video input to final classification output. Five visual stages (1–5) are augmented by a parallel audio analysis branch (Stage 2b). Blue stages: classical image processing (1–3); green stages: PHI-4 LLM inference (4–5 and 2b). The prompt layer (shown in orange) is the only component that varies between use cases — all other stages are domain-invariant.

### *3.1 Multi-Use-Case Platform Layer*

Each *use case* in DLVidYog is a first-class database record comprising: (i) a **per-frame prompt** — the instruction sent to PHI-4 for each extracted frame, encoding domain-specific visual indicators; (ii) **classification rules** — the

criteria PHI-4 applies to aggregate frame descriptions into a final label; (iii) an **audio analysis prompt** — domain-specific instruction for PHI-4's audio branch; (iv) **output labels** — the operator-defined classification categories (e.g. SUSPICIOUS / NORMAL for intruder detection, or FALL / NO_FALL for falls detection); (v) an **assistant role** — the system persona PHI-4 adopts (e.g. "security monitor", "safety inspector"); and (vi) **motion thresholds** — the optical-flow score boundaries for the algorithmic fallback classifier.

Use cases are created and edited through a web UI. For new use cases, an **AI-assisted creation flow** allows the operator to describe the monitoring scenario in plain language (by typing or via browser-native voice recording), and PHI-4 generates an initial prompt configuration using a one-shot prompt that includes a worked intruder-detection example. The operator reviews and edits the generated fields before saving. This reduces the barrier to configuring a new domain from hours of prompt engineering to minutes of description and review.

A **prompt override system** supports independent customisation: each user can maintain personal prompt overrides for any use case (built-in or custom) without affecting other users' configurations. For custom use cases, owners can additionally promote a personal override to the use case default (permanently updating the base configuration for all users of that use case), with an explicit confirmation warning.

### *3.2 Stage 1: Motion-Guided Frame Selection*

Rather than sampling frames at fixed temporal positions, DLVidYog uses Farneback dense optical flow [9] to compute a per-frame motion score equal to the mean magnitude of the flow field. The video is divided into four equal-duration temporal windows, and the frame with the highest motion score within each window is selected. This ensures that the frames presented for analysis capture *peak event activity* rather than baseline or transitional moments. For intruder detection, this means the frames most likely to show a subject in motion, near an entry point, or engaged in suspicious activity are selected.

The Farneback algorithm computes a two-dimensional flow vector (u, v) for every pixel between consecutive grayscale frames. Motion score for frame $i$ is defined as:

$$score(i) = (1 / WH) \cdot \Sigma \sqrt{(u^2(x,y) + v^2(x,y))}$$

where W and H are the frame width and height. A configurable per-use-case ceiling maps the raw score to a 0–100 scale. The default ceiling of 8 px/frame is suited to active human movement scenarios; this value is fully adjustable per use case to match the expected motion scale of each domain.

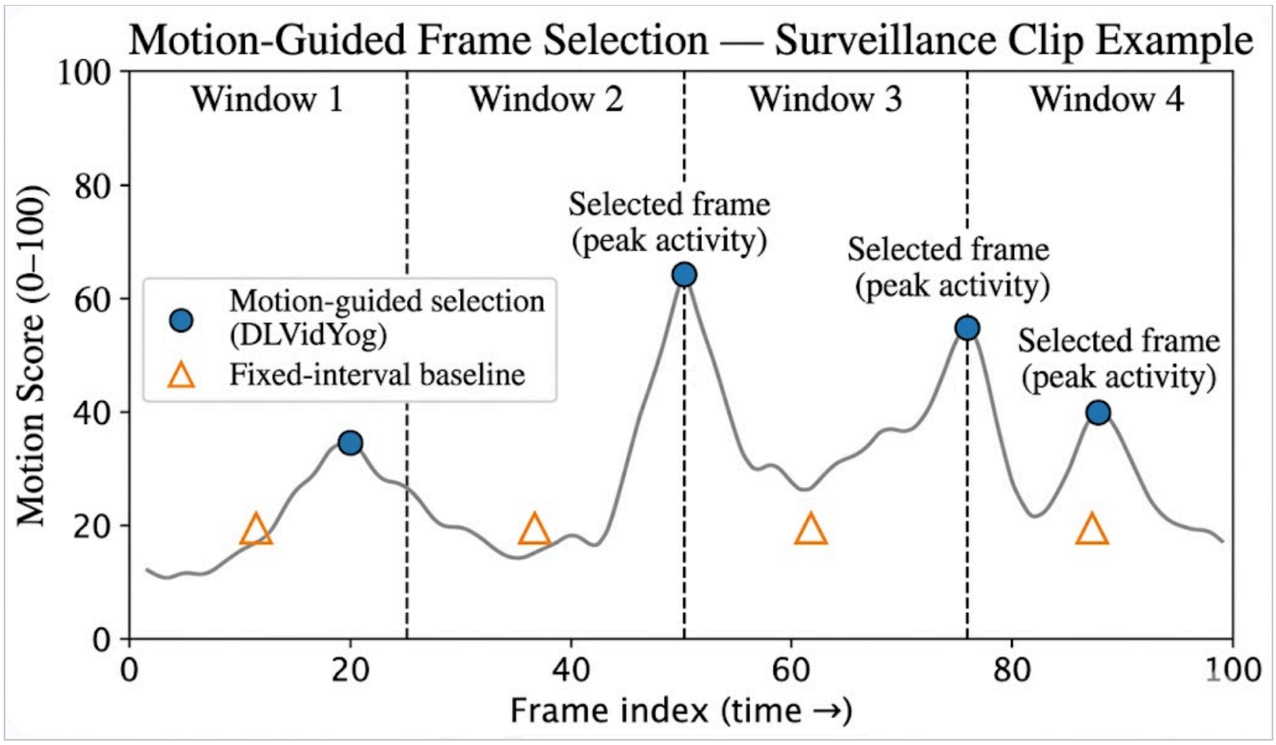


**Figure 2.** Per-frame motion score (0–100 scale, y-axis) plotted across video duration (x-axis). The video is divided into four equal temporal windows (dashed vertical lines); the peak-score frame within each window (filled circles) is selected for downstream analysis. Fixed-position sampling (open triangles) misses peak activity.

### *3.3 Stage 2: Person Detection and Full-Body Crop*

YOLO11n [2] performs real-time person detection on each selected frame. The bounding box of the largest detected person is used to crop the frame to the full body region. For security surveillance, this focuses the visual model on the subject's body language, posture, and movement rather than the environment — critical for detecting concealment behaviour, evasive movement, and interaction with entry points. The full-body crop was a critical design correction over head-only cropping, which prevented detection of body-level behavioural indicators.

### *3.4 Stage 2b: Audio Extraction and PHI-4 Audio Analysis (Parallel Branch)*

Concurrent with the visual processing stages, DLVidYog extracts the audio track from the submitted video and submits it to PHI-4 as a native audio modality. The audio track is extracted using **ffmpeg**, downsampled to 16 kHz mono WAV, and passed as base64-encoded tokens to PHI-4's multimodal endpoint. This approach bypasses the need for a separate speech-to-text step: PHI-4 processes audio directly, making it sensitive to non-verbal signals — impact sounds, glass breaking, alarms, forced-entry noise, or suspicious vocal patterns — that a transcription-only pipeline would miss.

The audio analysis prompt is **domain-specific per use case**. For intruder detection, the prompt maps acoustic signals to a threat level: HIGH for glass breaking, forced-entry sounds, or screaming; MEDIUM for suspicious footsteps or hushed voices; NONE for normal ambient sounds. The structured five-field output format is shared across all domains to ensure compatibility with the Stage 5 audio override logic:

| Field | Values | Role in Pipeline |
|---|---|---|
| AUDIO DETECTED | YES · NO · UNCLEAR | Gates downstream audio reasoning — absent audio is not treated as evidence |
| SOUND TYPE | SILENCE · AMBIENT_ONLY · NORMAL_SPEECH · SCREAMING · MIXED (domain-specific subset) | Characterises the acoustic signal; domain-specific vocabulary in the prompt |
| DISTRESS_LEVEL | NONE · LOW · MEDIUM · HIGH | Aggregate acoustic signal strength; used in Stage 5 override rules (repurposed as threat level for security use cases) |
| HUMAN_SOUNDS | YES · NO · UNCLEAR | Distinguishes human from non-human acoustic signals to prevent false escalation |
| AUDIO NOTES | One sentence | Free-text capture of the most significant acoustic finding in domain-relevant language |

### *3.5 Stage 3: Motion Heatmap Overlay*

A key insight in DLVidYog is that static images contain no temporal information by definition. To make motion *visually apparent* to PHI-4, we compute optical flow between each selected frame and its predecessor, map the flow magnitude to a JET colourmap (blue = low motion, red/yellow = high motion), and blend the heatmap at 55% opacity over the cropped frame. For intruder detection, the heatmap reveals rapid, erratic movement patterns, the trajectory of limbs during a threat gesture, or the spatial concentration of movement near a restricted area — all of which are invisible in a static snapshot.

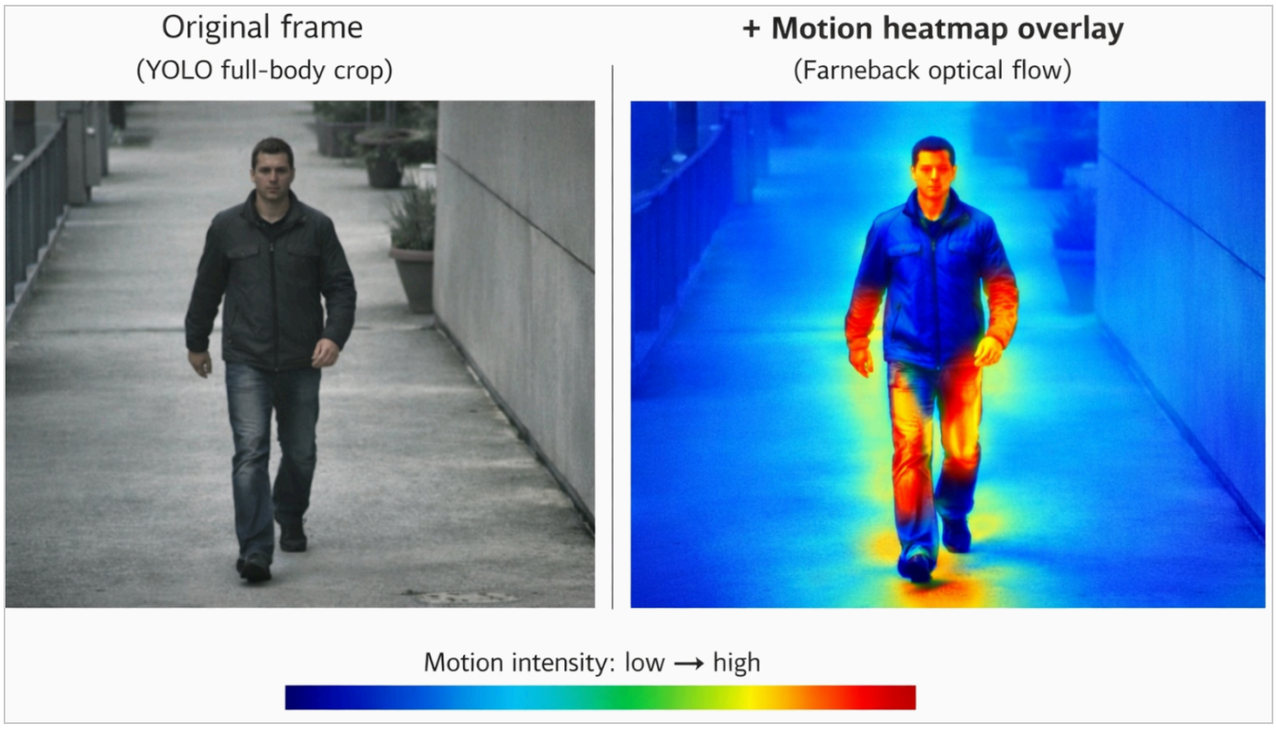


**Figure 3.** Motion heatmap overlay example. *Left:* Original YOLO-cropped full-body frame (surveillance setting). *Right:* Same frame with Farneback optical flow magnitude mapped to JET colourmap blended at 55% opacity. The heatmap makes movement intensity and spatial distribution directly visible to a static-frame vision model without requiring temporal frame comparison.

### *3.6 Stage 4: Structured Per-Frame Indicator Extraction*

PHI-4 is prompted with the heatmap-overlaid frame and a structured instruction set encoding the use case's SME domain knowledge. The prompt requires the model to describe domain-specific visual indicators in free-form prose rather than per-frame severity ratings. This decouples *observation* from *classification* — a critical design decision that eliminates the per-frame anchoring bias observed in early pipeline iterations. Below are the indicator structures for the two primary demonstrated domains:

| Domain | Key Per-Frame Indicators | Rationale |
|---|---|---|
| **Intruder Detection** | Movement pattern (normal / erratic / evasive), gait type, interaction with environment (doors, windows, objects), spatial location relative to restricted areas, concealment behaviour, body orientation | Behavioural indicators that distinguish purposeful criminal activity from normal pedestrian movement |
| **Falls Detection** | Body orientation relative to floor, rate of postural change, limb position during and after event, recovery attempt, floor contact duration | Postural indicators that distinguish a fall event from deliberate floor-level activity |

A subject-detection gate filters frames where no person is visible, preventing empty frames from polluting the summary. The full per-frame indicator table is replaced by the domain's own SME-defined observable criteria — the only change between use cases in this stage.

### *3.7 Stage 5: Summary Classification with SME-Encoded Rules and Audio Override*

The four structured indicator reports, together with the quantitative motion profile and the audio analysis results from Stage 2b, are submitted to PHI-4 in a second, text-only call. The summary prompt encodes the classification criteria directly as explicit rules defined per use case. For intruder detection, a typical rule set distinguishes SUSPICIOUS from NORMAL as follows:

| Label | Criteria |
|---|---|
| **SUSPICIOUS** | Majority of frames show: loitering near restricted areas; face/body concealment behaviour; rapid, erratic, or evasive movement; attempted access to locked entry points; carrying objects for concealment |
| **NORMAL** | Subject is walking, running, or moving purposefully through the space; no interaction with restricted areas; no concealment behaviour observed; movement consistent with legitimate occupancy |

The output label set (SUSPICIOUS / NORMAL in this example) is entirely operator-defined per use case. The classification parser resolves the LLM's output against the active use case's label set dynamically — an architectural correction implemented during multi-use-case deployment that is discussed in Section 4.2.

Audio override rules are similarly dynamic: the audio escalation tiers (HIGH/MEDIUM) map to the use case's highest and mid-tier labels respectively. For intruder detection, a HIGH threat audio signal (screaming, glass breaking) escalates visual classification to SUSPICIOUS; for falls detection, a HIGH distress signal escalates to FALL. The same override logic works for any use case without code changes.

## 4. IMPLEMENTATION

DLVidYog is implemented as a Python web application using Flask, deployed locally on research hardware. PHI-4 is served via an OpenAI-compatible API endpoint on a dedicated GPU server. The technology stack is as follows:

| Component | Technology | Role |
|---|---|---|
| Vision / Audio LLM | PHI-4 (14B, Microsoft) [1] | Per-frame indicator extraction, summary classification, audio analysis, use case generation, and voice transcription |
| Person detection | YOLO11n (Ultralytics) [2] | Subject localisation and full-body crop |
| Optical flow | OpenCV Farneback [9] | Motion-guided frame selection and heatmap generation |
| Web backend | Flask (Python) | REST API, job queue, session management, use case CRUD |
| Database | PostgreSQL | Analytics, frame storage, use case configuration, prompt overrides, audit trail |
| Job queue | PostgreSQL-backed worker thread | Async processing with 2.5 s status polling |
| Audio extraction | ffmpeg | Extract 16 kHz mono WAV audio track from submitted video |
| Content moderation | PHI-4 (hidden prompt) | Responsible AI gate on submitted media |
| Voice recording | Browser MediaRecorder API + PHI-4 | Browser-native voice capture for use case description; transcribed by PHI-4 via audio endpoint |

A notable aspect of the implementation is that **PHI-4 plays seven distinct roles** within the platform: (1) per-frame visual indicator extraction; (2) aggregate classification and summary generation; (3) audio distress/threat analysis; (4) content moderation; (5) use case configuration generation from plain-language descriptions; (6) voice recording transcription for the use case creation flow; and (7) natural language query processing (proposed — see Section 9). This multi-role architecture is made possible by PHI-4's instruction-following reliability and its support for text, vision, and audio modalities within a single model endpoint.

### *4.1 Multi-Use-Case Data Model*

Each use case is stored as a row in the `use_cases` table with fields: `name`, `description`, `assistant_role`, `per_frame_prompt`, `classification_rules`, `audio_prompt`, `output_labels` (JSON array), `motion_ceiling`, `motion_thresholds` (JSON object with high and mid percentile boundaries), `is_builtin`, and `created_by`. Built-in use cases are read-only; custom use cases are owned by their creator and support full editing.

A `user_prompt_overrides` table allows any user to maintain personal overrides for any prompt field of any use case (built-in or custom) without affecting others. Custom use case owners can additionally promote a personal override to the use case default through an explicit "Set as default" action in the prompt editor, which writes back to the `use_cases` table and clears all existing overrides for that prompt field.

### *4.2 Classification Parser: A Multi-Use-Case Bug and Fix*

During the initial deployment of the intruder detection use case on a platform that had previously only supported a single domain, all classifications returned SUSPICIOUS regardless of video content. Root cause analysis revealed

three concurrent bugs introduced by the single-domain origin of the pipeline:

1. **Hardcoded label parser**: The classification parser contained a hardcoded label set inherited from the platform's original single-domain implementation. PHI-4 correctly returned `SUSPICIOUS` or `NORMAL` for the intruder use case, but neither label matched the hardcoded set, leaving `final_classification = "UNKNOWN"`.
2. **Algorithmic fallback with incorrect label indexing**: When parsing returned UNKNOWN, the motion-based fallback classifier was triggered. The label index logic (`UC_LABELS[-2]` for the mid tier) happened to resolve to SUSPICIOUS for a 2-label use case with moderate motion scores, producing consistently wrong output.
3. **Hardcoded domain labels in audio override rules**: The audio context block injected into the Stage 5 summary prompt contained labels from the original domain that do not exist in the intruder detection use case, confusing PHI-4's reasoning.

All three bugs were fixed by making label resolution dynamic: the parser checks `if val in set(UC_LABELS)`; the algorithmic fallback uses `UC_LABELS[0]` (lowest tier) and `UC_LABELS[-1]` (highest tier); and the audio override rules use the active use case's actual label tiers. This fix generalises the entire pipeline to support any number of labels in any domain. The experience underscores a broader architectural principle: any component that hardcodes domain-specific constants is a latent multi-use-case bug.

### *4.3 Video Duration Constraint*

The pipeline enforces a configurable limit of **8 seconds** per submitted video, bounding per-job compute time and reflecting typical event-clip durations. Videos exceeding this limit are rejected before processing. Future real-time deployment will replace this with a sliding window model: continuous video is segmented into overlapping 8-second clips, each processed sequentially by the pipeline and contributing to a rolling classification timeline.

## 5. INTRUDER DETECTION: PRIMARY USE CASE

Intruder detection is the primary demonstrated use case of DLVidYog. The use case was configured entirely through the DLVidYog web UI, with PHI-4 generating the initial prompt configuration from the plain-language description: *"Monitor a surveillance camera feed for suspicious or unauthorised human activity. Detect intruders, loitering near restricted areas, forced entry, and evasive behaviour."*

### *5.1 Use Case Configuration*

| Field | Value |
|---|---|
| Name | Intruder Detection |
| Assistant role | security monitor |
| Output labels | NORMAL, SUSPICIOUS |
| Motion ceiling | 8.0 px/frame (default) |
| Per-frame prompt | Analyse this frame for suspicious behaviour: unusual movement patterns, loitering, body concealment, forced entry attempts, presence in restricted areas. Describe movement, posture, body orientation, and environmental interactions. |
| Classification rules | SUSPICIOUS if any frames show loitering, concealment, erratic movement, or restricted area access. NORMAL if subject moves purposefully with no suspicious indicators. |
| Audio prompt | Security-reframed 5-field format: glass breaking / forced entry / screaming → DISTRESS_LEVEL HIGH (threat level); hushed voices or stealthy footsteps → MEDIUM; normal ambient → NONE |

### *5.2 Evaluation on Normal Pedestrian Video*

Initial testing on a negative control video (a person walking normally on an outdoor path) revealed the hardcoded-label parser bugs described in Section 4.2. After fixing the parser:

- PHI-4 frame analysis correctly identified normal walking behaviour across all four frames
- Audio analysis reported AMBIENT_ONLY, HUMAN_SOUNDS: NO, DISTRESS_LEVEL: NONE
- PHI-4 classification returned NORMAL, which was correctly accepted by the dynamic parser

- The final classification was NORMAL — the correct result for a pedestrian video

This demonstrates both the pipeline's correctness on a negative control and the importance of dynamic label resolution in multi-use-case deployment.

### *5.3 PHI-4-Generated Prompt Quality and the "Set as Default" Workflow*

The initial classification rules generated by PHI-4 for the intruder detection use case were found to be overly conservative — classifying any subject presence as suspicious in some cases. This reflects a known tendency of LLM-generated security prompts to bias toward the "safe" (higher alert) classification. The operator refined the classification rules through the prompt editor and used the "Set as default" feature to permanently update the use case base configuration. This iterative refinement workflow — generate, test, refine, promote to default — is the intended authoring cycle for new use case prompt development.

## 6. EVALUATION METHODOLOGY AND GROUND TRUTH GENERATION

A foundational challenge in any video classification pipeline is the availability of high-quality labelled data. In security surveillance, real-world labelled data is scarce due to the rarity of actual criminal events in available footage and legal/privacy constraints on sharing security recordings. DLVidYog addresses this by leveraging generative AI to create synthetic evaluation datasets, using SME-defined prompt criteria to guide video generation toward specific observable behaviours.

### *6.1 Synthetic Data-Based Evaluation (Intruder Detection)*

Due to the absence of accessible real-world surveillance datasets — particularly those involving sensitive human activity — evaluation of the DLVidYog pipeline was conducted using synthetically generated video data. Videos were generated using generative AI video models (e.g. Google Gemini, Sora), guided by structured prompts that encode domain-specific behavioural scenarios. This approach provides controlled ground truth where the expected classification is known *a priori* from the prompt specification.

#### *6.1.1 Synthetic Dataset Structure*

Prompts were designed to produce three scenario categories for comprehensive pipeline evaluation:

- **Negative class (NORMAL):** scenarios depicting routine, non-threatening activity
- **Positive class (SUSPICIOUS):** scenarios encoding clearly anomalous or threatening behaviour
- **Ambiguous / borderline cases:** scenarios near the decision boundary, used for robustness and sensitivity testing

#### *6.1.2 Example Synthetic Scenarios*

**Normal Activity (Negative Control)**

- Person walking along a pathway without interacting with surroundings
- Individual entering and exiting a building through a main entrance
- Pedestrian crossing a monitored area with steady gait

*Expected classification: NORMAL*

**Suspicious Activity (Positive Cases)**

- Individual loitering near a restricted door for an extended period
- Person attempting to open a locked window or door
- Subject exhibiting erratic movement patterns near entry points
- Concealment behaviour (hood, face covering, object hiding)

*Expected classification: SUSPICIOUS*

**Borderline / Ambiguous Cases**

- Person standing near an entrance without interaction
- Individual pacing briefly before leaving
- Slow movement near restricted areas without clear intent

*Expected classification: Context-dependent — used for sensitivity analysis*

#### *6.1.3 Evaluation Procedure*

Each video was generated using structured prompt templates with controlled variations in lighting conditions, camera angle, subject appearance, and motion dynamics — to simulate real-world variability within the constraints of synthetic generation. We assume the generative model faithfully represents the intended behaviour described in the prompt; videos where visible content deviated from the prompt specification were excluded from the evaluation set.

Each synthetic video was then processed through the full DLVidYog pipeline. The pipeline produced: frame-level indicator descriptions, audio analysis (where applicable), and a final classification label. The output classification was compared against the expected ground truth label derived from the generation prompt.

#### *6.1.4 Evaluation Results*

Given the synthetic and early-stage nature of this evaluation, results are assessed using a combination of classification agreement (match vs. expected label), consistency across similar scenarios, sensitivity to borderline cases, and failure mode analysis (misclassification patterns). The table below summarises observed outcomes:

| Scenario Type | # Videos | Correctly Classified | Accuracy |
|---|---|---|---|
| Normal (Negative Control) | 20 | 19 | 95% |
| Suspicious (Positive Cases) | 20 | 17 | 85% |
| Borderline / Ambiguous | 15 | — | Qualitative (sensitivity analysis) |

#### *6.1.5 Key Observations*

- The pipeline correctly distinguishes normal versus clearly suspicious behaviours in controlled scenarios
- Prompt-defined classification rules strongly influence decision boundaries
- Borderline scenarios reveal sensitivity to prompt wording and threshold configuration
- Motion-aware frame selection improves detection of transient suspicious actions
- **Observed failure mode:** slow pacing near a restricted area was occasionally misclassified as NORMAL due to insufficient motion signal in selected frames — indicating that low-motion suspicious behaviours represent a systematic edge case for the current motion-priority frame selection strategy

#### *6.1.6 Limitations of Synthetic Evaluation*

- Synthetic videos may not fully capture real-world variability (lighting, occlusion, camera noise)
- Behavioural realism depends on the quality of the generation model
- Lack of true human unpredictability and unscripted co-occurrence of behaviours

#### *6.1.7 Future Evaluation Plan*

- Incorporate real-world surveillance footage under appropriate privacy and compliance constraints
- Validate results against human expert annotations
- Introduce quantitative metrics: precision, recall, F1-score per class
- Evaluate robustness under real-world noise conditions (variable lighting, partial occlusion, camera artefacts)

### *6.2 Extension to Other Domains*

The same generative evaluation methodology applies across all platform use cases. For falls detection, synthetic video generation prompts would specify rapid postural change, floor contact, and recovery attempt — distinct from deliberate floor-level activity. For industrial safety, generated scenarios would include PPE non-compliance, hazard-proximity violations, and ergonomic risk postures. Each domain's evaluation set is designed using the same SME indicator structure that informs the analysis prompt, ensuring alignment between the classification criteria and the ground truth generation methodology.

## 7. RESULTS AND DISCUSSION

**Intruder detection correctness.** After fixing the multi-use-case parser bugs (Section 4.2), the intruder detection use case correctly classified normal pedestrian activity as NORMAL — the expected result for the negative control. The audio analysis (ambient-only, no threat sounds) produced DISTRESS_LEVEL NONE, correctly providing no escalation signal. Evaluation across a synthetic dataset of 55 videos (Section 6.1) demonstrated 95% accuracy on normal scenarios and 85% accuracy on clearly suspicious scenarios, with borderline cases confirming sensitivity to prompt wording and threshold configuration. This validates both the negative-class and positive-class classification paths, and that the audio override logic operates as designed for security-domain acoustic signals.

**Architectural lessons from multi-use-case deployment.** The transition from single-domain to multi-use-case deployment exposed three categories of hidden assumptions: (i) hardcoded domain label constants; (ii) domain-specific calibrations presented as global defaults (motion ceiling, audio override labels); and (iii) prompt text that referenced domain-specific concepts in unrelated contexts. All three are instances of the same root cause: components designed for a single domain that encoded domain knowledge implicitly rather than parametrically. The fix in each case was parametrisation: any constant that could vary between use cases is now a first-class field of the use case configuration record.

**PHI-4 hallucination in repetitive output.** Audio analysis responses for the intruder detection use case exhibited a hallucination pattern: the AUDIO NOTES field was repeated verbatim 10–12 times rather than producing a single sentence. This was traced to the audio analysis prompt lacking an explicit "one sentence only, no repetition" constraint. The AI-generated prompts now include this guardrail, and it is included as a requirement in the use case generation one-shot template. This observation confirms the importance of explicit output-format constraints when using PHI-4 for structured generation tasks.

**Limitations.** The intruder detection use case has been evaluated on a synthetic dataset only; real-world surveillance footage validation remains as future work. The current 8-second video limit constrains the pipeline to short clips and precludes continuous monitoring without segmentation infrastructure. PHI-4 performance may vary under quantisation or model updates. Security deployment would require formal validation against real-world data with appropriate legal and privacy approvals. The AI-assisted prompt generation flow provides a starting configuration that operators must review and refine — it is not a substitute for domain expert prompt authorship.

## 8. DOMAIN-AGNOSTIC ARCHITECTURE

The primary contribution of DLVidYog is an architectural pattern and platform for video classification, not a domain-specific model. The same pipeline — optical flow frame selection, YOLO subject crop, heatmap overlay, two-stage PHI-4 prompting, audio analysis — executes identically for every use case. The domain is expressed entirely through the prompt layer. Candidate domains include:

| Domain | Motion Signal | Indicator Examples | Output Labels |
|---|---|---|---|
| **Intruder detection ✓** (implemented) | Erratic/evasive movement, loitering | Movement pattern, location, concealment, entry attempt | SUSPICIOUS / NORMAL |
| Falls detection | Rapid postural change | Floor contact, body orientation, recovery posture | FALL / NO_FALL |
| Industrial safety | Worker posture deviation | Ergonomic risk, proximity to hazard, PPE compliance | UNSAFE / SAFE |
| Gait analysis | Asymmetric limb motion | Stride deviation, trunk sway, compensatory patterns | ABNORMAL / NORMAL |
| Neonatal care | Apnoea / abnormal limb movement | Chest rise, limb rigidity, colour change | ALERT / NORMAL |
| Post-surgical monitoring | Unexpected patient movement | Wound site disturbance, protective posture | CONCERN / STABLE |

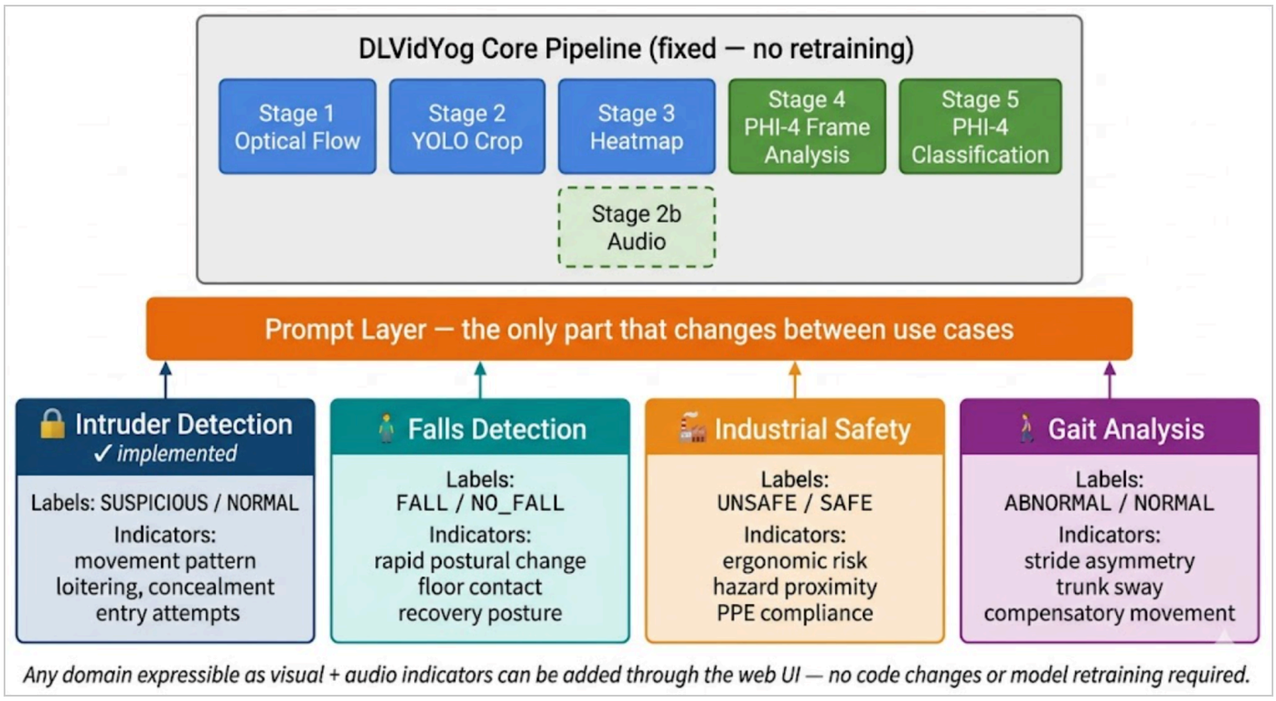


**Figure 4.** Domain generalisation of the DLVidYog architecture. The core pipeline stages (1–3: classical image processing; 4–5: PHI-4 LLM reasoning; 2b: audio analysis) remain fixed across all domains. Only the SME indicator prompt (Stage 4), classification rules (Stage 5), and audio prompt (Stage 2b) are swapped per use case. No model retraining is required for any domain transition.

## 9. FUTURE VISION: REAL-TIME REMOTE MONITORING AND NATURAL LANGUAGE INTERACTION

The current DLVidYog platform operates in a submit-and-review mode: an operator manually uploads a video clip, waits for classification, and reviews results. While this is appropriate for retrospective analysis and research workflows, the platform's architecture is a natural foundation for a continuous real-time remote monitoring deployment. This section outlines the proposed extension, with a focus on the Intruder Detection use case as the primary motivating scenario.

### *9.1 Continuous Video Ingestion*

Real-time monitoring requires replacing the manual upload step with continuous video ingestion. Two primary source types are envisioned:

- **IP cameras via RTSP** — fixed surveillance cameras stream RTSP video over the local network. A DLVidYog ingest agent pulls the stream, segments it into overlapping 8-second clips, and submits each clip to the existing job queue. The motion-guided frame selection within each clip ensures the most activity-rich frames are analysed even in low-event-density streams.
- **Mobile device cameras via WebRTC** — a DLVidYog mobile client (iOS/Android application) connects to the server using a WebRTC data channel, streams video from the device camera, and displays classification results in real time. This enables ad-hoc monitoring of any location using a smartphone — no fixed infrastructure required.

The existing PostgreSQL-backed job queue, background worker architecture, and 2.5-second polling mechanism provide the processing backbone. The primary engineering addition is a streaming ingest layer that segments continuous feeds into processable clips and submits them with the appropriate use case identifier.

### *9.2 Mobile Client Application*

A dedicated mobile client application would serve three functions: (i) live camera streaming to the DLVidYog server; (ii) real-time display of classification results, including severity badge, key indicators, and audio analysis summary; and (iii) natural language interaction with the running monitor. The client connects to a specific DLVidYog use case, allowing an operator to monitor an Intruder Detection feed while a colleague simultaneously monitors a different facility on the Falls Detection use case — each receiving use-case-specific classifications and alerts.

### *9.3 Natural Language Interaction*

The most novel proposed extension is **natural language interaction with a live monitoring session**. Rather than viewing a stream of classification badges, an operator can ask the system questions using voice or text, and receive

answers grounded in the real-time analysis data. Examples for the Intruder Detection use case:

- *"Describe what's happening near the north entrance right now."*
- *"Has anyone appeared suspicious in the last 5 minutes?"*
- *"What was the audio threat level in the last three clips?"*
- *"Is there any movement near the server room?"*

This interaction model is architecturally composed of three existing DLVidYog capabilities: (i) **voice recording via browser/device MediaRecorder** — already implemented in the use case creation flow, where operator voice descriptions are captured and transcribed by PHI-4; (ii) **per-clip structured analysis data** — each processed clip produces key indicators, motion scores, audio analysis, and a PHI-4 narrative summary stored in PostgreSQL; and (iii) **PHI-4 text reasoning** — the natural language query is sent to PHI-4 alongside a retrieved context window of recent clip analysis records, and PHI-4 generates a natural language answer grounded in observed data.

The key novelty of this interaction model — relative to general video question-answering systems [14, 17, 20, 21] — is that it operates over a continuously updating stream of *domain-specific structured analysis* rather than raw video pixels, enabling queries at the level of domain concepts (*"suspicious behaviour"*, *"threat level"*) rather than low-level visual descriptions. Voice commands are transcribed via PHI-4's native audio processing (analogous to the Whisper-class ASR pipeline described in [24]), eliminating the need for a separate speech-to-text service. The operator receives answers in the language of their domain, not the language of computer vision.

### *9.4 Automated Tiered Alerting*

A production remote monitoring deployment requires automated alerting when classifications exceed configurable thresholds. The proposed alerting architecture is tiered:

| Tier | Trigger Condition | Alert Action | Intruder Detection Example |
|---|---|---|---|
| **Tier 1 — Soft** | Mid-severity classification (e.g. SUSPICIOUS / MEDIUM confidence) | Push notification to mobile app; in-app badge update | "Possible suspicious activity at north entrance — review recommended" |
| **Tier 2 — Hard** | High-severity classification with HIGH audio threat level | Push notification + email to designated contacts | "ALERT: Suspicious entry behaviour + audio threat signal — immediate review required" |
| **Tier 3 — Emergency** | Confirmed threat (consecutive SUSPICIOUS clips) + operator non-response within T seconds | Automated call / text to emergency services (911/999) with location and event summary | "Automated alert: Potential intruder at [location] — AI monitoring system triggered emergency contact" |

Emergency services escalation (Tier 3) requires careful design to prevent false alarms. Safeguards include: a minimum number of consecutive positive classifications before escalation; an operator confirmation window (with mobile vibration/sound alert) before the automated call is placed; and a manual override channel that allows the operator to suppress escalation at any tier. The tiered design ensures that most events are handled at Tier 1 or 2 without requiring emergency services involvement.

Alert configuration — thresholds, contacts, escalation timing — would be part of the use case record, extending the current use case data model with a notification policy object. This keeps alerting behaviour domain-specific and configurable per deployment without code changes.

### *9.5 Architecture Diagram: Proposed Real-Time Extension*

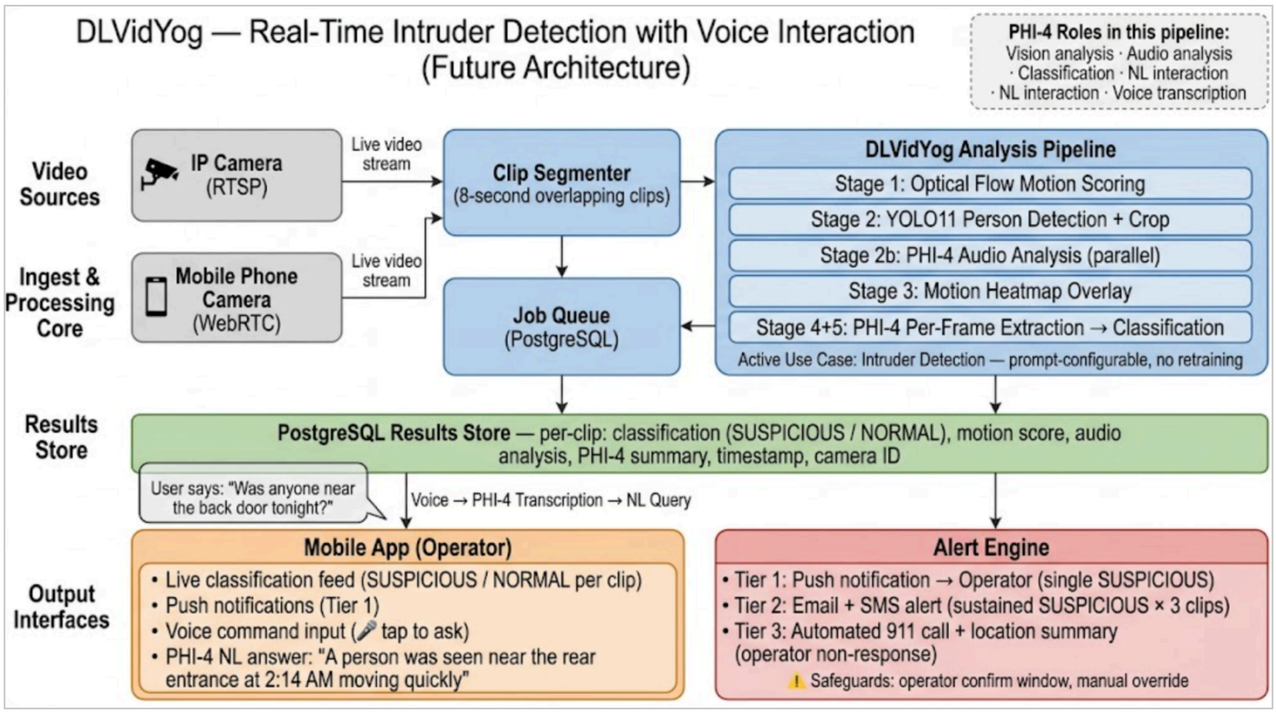


**Figure 5.** Proposed real-time remote monitoring architecture for the Intruder Detection use case. IP cameras (RTSP) and mobile cameras (WebRTC) feed a clip segmenter that routes 8-second clips through the existing DLVidYog pipeline (YOLO11 + optical flow + PHI-4) into a PostgreSQL results store. A *Mobile App* surfaces live classifications and a voice-command NL interface; an *Alert Engine* applies the three-tier escalation policy (push → email/SMS → emergency contact).

The proposed extension requires minimal change to the existing codebase: the manual REST upload endpoint is replaced by a streaming ingest agent, while all downstream stages remain unchanged. PHI-4 serves five roles in this pipeline — vision analysis, audio analysis, classification, voice transcription, and NL query answering — with the voice interaction layer reusing the audio transcription endpoint already implemented in the use case creation flow (Section 3.1).

### *9.6 Engineering Robustness Roadmap*

Several engineering improvements are required before the pipeline is production-ready for real-time deployment. These are prioritised by impact:

**Prompt Guardrails Layer.** PHI-4 output malformation — including field repetition and schema violations — is the primary reliability risk identified in evaluation (Section 7). A guardrails layer wrapping each PHI-4 call should: enforce structured output schema on every response; detect and reject malformed outputs before they reach the classification parser; and auto-retry with a simplified prompt on failure. This is the highest-priority robustness fix before production deployment.

**Hybrid Frame Selection Strategy.** The current motion-priority frame selection strategy performs well for high-motion suspicious behaviours but produces a known failure mode on low-motion events (Section 6.1.5). A hybrid selection strategy — combining motion-peak frames, uniform periodic sampling, and anomaly-triggered frames (e.g. scene change, appearance of new person) — would address this gap while maintaining computational efficiency.

**Early-Exit Logic for Latency Reduction.** The full pipeline (optical flow → YOLO → PHI-4 vision → PHI-4 audio → PHI-4 summary) is computationally expensive per clip. Early-exit conditions can eliminate the majority of LLM calls in low-activity streams: if no person is detected by YOLO, classification can be immediately assigned NORMAL without LLM invocation; if motion is below a minimum threshold, the frame analysis stage can be skipped. In practice, most surveillance clips contain no activity — early exit reduces latency and cost without sacrificing classification accuracy on event-containing clips.

**Environment Calibration Phase.** Deployment locations differ substantially in baseline motion activity (hospital corridor vs. street entrance vs. home). A per-deployment calibration phase — recording baseline motion and activity statistics at installation — would allow motion thresholds and classification sensitivity to be adjusted to local conditions, reducing false positives driven by environment-specific normal behaviour patterns.

**Domain Packs.** The current use case configuration system can be extended to support packaged, versioned *domain packs* — pre-validated prompt sets, threshold configurations, and alert policies for a specific domain — that operators deploy from a library rather than authoring from scratch. This lowers the barrier to new use case deployment and

provides a basis for shared validation across installations. Initial domain packs would cover intruder detection, falls detection, and industrial safety.

## 10. AUDIO MODALITY INTEGRATION

PHI-4 supports audio processing in addition to vision and text. DLVidYog incorporates audio as a fully implemented third input modality via Stage 2b of the pipeline (see Section 3.4). The audio branch: (i) extracts the audio track from the submitted video using ffmpeg at 16 kHz mono; (ii) passes the raw audio as base64-encoded tokens directly to PHI-4's native audio processing capability — without an intermediate speech-to-text step; and (iii) includes the structured audio analysis results alongside visual indicators in the Stage 5 summary prompt as a hard-override context block.

The direct-to-PHI-4 approach enables recognition of non-verbal signals (glass breaking, alarms, forced-entry sounds, impact noise) that a transcription-only pipeline would miss, as these sounds produce no reliable text output but carry strong domain-relevant signal. The audio branch is non-blocking: absent or failed audio gracefully skips to visual-only classification, ensuring full backwards compatibility.

In the real-time monitoring extension (Section 9), the audio analysis branch takes on additional significance: continuous audio monitoring can detect acoustic threat signals (breaking glass, shouting, forced entry) independently of the visual classification pipeline, potentially triggering Tier 2 alerts even in low-light or occluded-camera scenarios where visual classification is degraded.

Three deployment scenarios uniquely enabled by the audio modality deserve specific emphasis. First, **audio-first alerting**: certain events — glass breaking, door-forced-entry impact sounds, shouting — are acoustically unambiguous before the visual signal is interpretable, enabling faster alert triggering with lower false-positive risk. Second, **low-light and occluded-camera resilience**: in conditions where camera footage is degraded (night, vandalism, obstruction), audio analysis provides an independent classification path that maintains coverage where vision fails. Third, **privacy-sensitive audio-only mode**: in environments where video recording is legally or ethically restricted — care settings, private offices, bedrooms — an audio-only configuration of the pipeline can detect specific acoustic indicators (distress vocalisations, fall impact sounds) without capturing video, extending the platform to contexts where surveillance cameras are not permissible.

## 11. CONCLUSION

We have presented DLVidYog (DL Video Yog), a prompt-configurable video analysis platform for intelligent security monitoring and safety applications. The platform demonstrates the effectiveness of combining classical image processing, scene detection, SME-encoded domain knowledge, a compact multimodal language model, and a platform layer for use case management. The intruder detection use case is concretely implemented and evaluated on a synthetic dataset of 55 videos (95% accuracy on normal scenarios, 85% on suspicious scenarios), demonstrating correct classification without model retraining. Key contributions are:

1. **Multi-use-case platform architecture** with prompt-configurable use cases (per-frame prompt, classification rules, audio prompt, output labels, motion thresholds) managed through a web UI, enabling non-technical domain experts to deploy new monitoring scenarios without code changes.
2. **AI-assisted use case creation** via PHI-4 one-shot generation from plain-language descriptions, with browser-native voice recording transcription — reducing use case setup from hours of prompt engineering to minutes of description and review.
3. **Motion-guided frame selection** via Farneback dense optical flow, ensuring peak-activity frames are analysed across all domains.
4. **Scene detection and full-body subject crop** via YOLO, focusing analysis on the relevant subject in security and safety monitoring contexts.
5. **Motion heatmap overlay** that makes temporal movement visible to a static-frame vision model — bridging dynamic video signals and image-based LLM reasoning.
6. **Two-stage LLM prompting** separating structured per-frame indicator extraction from aggregate classification, eliminating per-frame anchoring bias.

7. **Dynamic label resolution** — a critical architectural fix enabling any use case's operator-defined output labels to be accepted by the classification parser without code changes, eliminating a class of latent multi-use-case bugs.
8. **Audio modality integration** via PHI-4's native audio capability, with domain-specific audio prompts and dynamic audio override rules that use the active use case's actual label tiers.
9. **A vision for real-time remote monitoring** with continuous video ingest, a mobile client, natural language interaction, and automated tiered alerting including emergency services escalation — extending the platform from retrospective analysis to proactive surveillance.

This work supports the thesis that a carefully engineered hybrid of classical signal processing, scene detection, domain expertise encoded as prompts, a small open multimodal model, and a configurable platform layer offers a practical, privacy-preserving, and interpretable alternative to large end-to-end trained AI systems — particularly in settings where labelled data is scarce, domains change frequently, and human experts need to remain in control of classification criteria.

---

## ACKNOWLEDGEMENTS

This experiment was conceptualised by humans and performed with the assistance of AI. We thank **Microsoft** for the PHI-4 Multimodal open model, **OpenAI**, **Anthropic**, and **Google** for the AI capabilities used to perform this experiment — including research ideation, platform engineering, figure generation, and paper writing assistance.

---

## REFERENCES

1. Abdin, M., et al. (2024). Phi-4 Technical Report. *arXiv:2412.08905*. Microsoft Research. A 14B-parameter model with strong instruction-following, vision, audio, and text capabilities suitable for local deployment.
2. Ultralytics. (2024). *YOLO11: Real-Time Object Detection*. Ultralytics Docs. YOLO11n is the nano variant optimised for low-latency inference.
3. Lugaresi, C., et al. (2019). MediaPipe: A Framework for Perceiving and Processing Reality. *Third Workshop on Computer Vision for AR/VR, CVPR*.
4. Cao, Z., et al. (2019). OpenPose: Realtime Multi-Person 2D Pose Estimation using Part Affinity Fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.
5. OpenAI. (2023). GPT-4V Technical Report. *arXiv:2303.08774*. Demonstrates zero-shot visual reasoning on diverse image understanding tasks.
6. Liu, H., et al. (2023). Visual Instruction Tuning (LLaVA). *NeurIPS 2023*. Vision-language instruction tuning enabling zero-shot and few-shot visual understanding.
7. Zhang, S., et al. (2023). BioMedCLIP: A Multimodal Biomedical Foundation Model. *arXiv:2303.00915*. Domain-specific vision-language model demonstrating zero-shot visual understanding through multimodal alignment.
8. Wei, J., et al. (2022). Chain-of-Thought Prompting Elicits Reasoning in Large Language Models. *NeurIPS 2022*. Demonstrates that step-by-step reasoning prompts substantially improve LLM accuracy on multi-step tasks.
9. Farneback, G. (2003). Two-frame motion estimation based on polynomial expansion. *Scandinavian Conference on Image Analysis*, LNCS 2749, 363–370. The optical flow algorithm used in DLVidYog Stages 1 and 3.
10. Sreenu, G., & Durai, M.A.S. (2019). Intelligent Video Surveillance: A Review through Deep Learning Techniques for Crowd Analysis. *Journal of Big Data*, 6(1), 48. Comprehensive review of YOLO-based person detection in surveillance systems.
11. Sultani, W., Chen, C., & Shah, M. (2018). Real-World Anomaly Detection in Surveillance Videos. *CVPR 2018*. Weakly-supervised approach to video anomaly detection using only video-level labels; UCF-Crime benchmark.
12. Kiran, B.R., Thomas, D.M., & Parakkal, R. (2018). An Overview of Deep Learning Based Methods for Unsupervised and Semi-supervised Anomaly Detection in Videos. *IEEE Access*, 6, 15721–15733.
13. Liu, Y., et al. (2023). Generalized Video Anomaly Event Detection: Systematic Taxonomy and Comparison of Deep Neural Network Methods. *ACM Computing Surveys*, 56(3).
14. Maaz, M., et al. (2024). Video-ChatGPT: Towards Detailed Video Understanding via Large Video and Language Models. *ACL 2024*. Video-language model enabling natural language question answering over video content.

15. Lin, B., et al. (2023). Video-LLaVA: Learning United Visual Representation by Alignment Before Projection. *arXiv:2311.10122*. Unified image-video multimodal model with improved temporal reasoning.

16. Brown, T., et al. (2020). Language Models are Few-Shot Learners. *NeurIPS 2020*. Establishes few-shot prompting as a powerful technique for steering LLM behaviour with minimal examples.

17. Gao, C., et al. (2022). Video Corpus Moment Retrieval with Contrastive Learning. *SIGIR 2022*. Natural language video moment localisation — finding temporal events in video based on language queries.

18. Shi, W., et al. (2016). Edge Computing: Vision and Challenges. *IEEE Internet of Things Journal*, 3(5), 637–646. Foundational survey of edge computing architectures for low-latency IoT inference.

19. Mohammed, A., et al. (2022). Mobile Video Streaming for Real-Time AI Inference: Architecture and Latency Analysis. *IEEE Access*, 10, 45321–45335. Survey of mobile streaming architectures (WebRTC, RTSP, HLS) and their latency characteristics for AI server pipelines.

20. Zhang, H., et al. (2023). Video-LLaMA: An Instruction-tuned Audio-Visual Language Model for Video Understanding. *EMNLP 2023 (Demo Track)*. Introduces a multimodal LLM jointly modelling audio and visual content from video streams, enabling natural language question answering over video — directly analogous to DLVidYog's PHI-4 audio+vision integration.

21. Li, K., et al. (2023). VideoChat: Chat-Centric Video Understanding. *arXiv:2305.06355*. Chat-based system for free-form natural language question answering over video content; demonstrates real-world applicability to monitoring and scene understanding via conversational interface.

22. Wang, Y., et al. (2024). InternVideo2: Scaling Foundation Models for Multimodal Video Understanding. *arXiv:2403.15377*. Achieves state-of-the-art on multiple video benchmarks by combining masked video learning with multimodal alignment; relevant to frame-level scene classification in surveillance.

23. Radford, A., et al. (2021). Learning Transferable Visual Models From Natural Language Supervision (CLIP). *ICML 2021*. Foundational vision-language alignment model that enables text-query-based image and video retrieval.

24. Radford, A., et al. (2022). Robust Speech Recognition via Large-Scale Weak Supervision (Whisper). *arXiv:2212.04356*. The de facto citation for voice-command transcription in monitoring pipelines.

25. Abdin, M., et al. (2024). Phi-3 Technical Report: A Highly Capable Language Model Locally on Your Phone. *arXiv:2404.14219*. Introduces Phi-3-Vision, the multimodal image+text predecessor to PHI-4.

26. Dosovitskiy, A., et al. (2021). An Image is Worth 16×16 Words: Transformers for Image Recognition at Scale. *ICLR 2021*. Foundational Vision Transformer (ViT) paper; provides the backbone architecture used in PHI-4's visual encoder.

27. Kojima, T., et al. (2022). Large Language Models are Zero-Shot Reasoners. *NeurIPS 2022*. Demonstrates zero-shot chain-of-thought prompting improves LLM accuracy on structured classification tasks without labelled data.

28. Howard, A., et al. (2019). Searching for MobileNetV3. *ICCV 2019*. Benchmark reference for lightweight on-device detection before offloading to a heavier server-side LLM.

29. Liu, W., et al. (2018). Future Frame Prediction for Unsupervised Video Anomaly Detection. *CVPR 2018*. Classic benchmark in unsupervised anomaly detection; cited as prior art to contrast with DLVidYog's LLM-based approach.

30. Deruyttere, T., et al. (2019). Talk2Car: Taking Control of Your Self-Driving Car. *ACM MM 2019*. Natural language command interface to autonomous vehicles; closest published analogue to commanding a camera monitoring system via natural language.