GLOBAL JOURNAL
OF COMPUTER SCIENCE AND TECHNOLOGY - D: NEURAL & AI
Volume 26 | 2026 Article ID: GJCST-259174 Language: English

Vision Language Models as the Reasoning Layer of an AI-CCTV Planning Toolkit: A Case Study of the IndoAI's Calculator Suite

Online First

Vivek Gujar[§ϑ*]
CSO
*Corresponding Author

Advait Gujar[‡ξ]
Intern

Ashwani Kumar Rathore[§θ]
CEO

§ IndoAI Technologies Pvt. Ltd., Pune, India
‡ Manipal Institute of Technology, Manipal, India

RECEIVED	ACCEPTED	ONLINE PUBLISHED	PEER REVIEW
2026-07-10	2026-07-20	2026-08-01	Double Blind

Abstract

Designing AI-enabled video surveillance systems has become far more challenging than simply deciding the number of cameras or estimating storage. Once AI inference is performed on cameras or edge devices, every design decision influences several others. For example, increasing the pixel density needed for facial recognition affects lens selection, bitrate, storage capacity, and network bandwidth, while adding more AI analytics changes edge-computing requirements and software licensing. As a result, conventional CCTV planning methods are no longer sufficient for modern AI deployments. This paper presents the architecture of IndoAI's integrated planning toolkit, which follows a three-stage workflow ie Design, Size, and Verify, using nine specialised calculators for camera placement, infrastructure sizing, and deployment validation. Alongside these tools, an Appization-based AI Agent License Sizing and ROI Predictor estimates software licensing requirements and deployment economics. Building on recent advances in Vision Language Models (VLMs) and platform economics, we propose a VLM as the natural-language interface to this planning framework. Users can describe surveillance requirements using text, photographs or floor plans instead of manually entering technical parameters. The VLM converts these inputs into structured data for the calculator suite. We also present two systems under development: a VLM-driven video intelligence pipeline that analyses surveillance footage to generate timestamped event detections, evaluated using the UCF-Crime dataset and a VLM-assisted Bill of Quantities generator that produces annotated camera layouts together with storage, bandwidth and licensing estimates for deployment planning.

Vision Language Models | AI video surveillance | Bill of Quantities automation | edge computing | platform economics | pixel density planning | license/agent provisioning

AI USE STATEMENT
No generative AI was used for analysis or results.

FUNDING
No external funding was declared for this work.

CONFLICT OF INTEREST
The authors declare no conflict of interest.

DATA AVAILABILITY
Not applicable for this article.

ETHICS
No ethics committee approval was required for this article type.

CONSENT
Not applicable for this article.

TRIAL REG.
Not applicable.

Crossref DOI: 10.34257/GJCSTD259174

How to Cite: Gujar et al. (2026). Vision Language Models as the Reasoning Layer of an AI-CCTV Planning Toolkit: A Case Study of the IndoAI's Calculator Suite. Global Journal of Computer Science and Technology, Ahead of Print, 1-15. DOI: 10.34257/GJCSTD259174

AR Experience Web Page DOI

Print ISSN 0975-4350
Online ISSN 0975-4172

Under the strict compliance and defined process of

METADATA CONTINUATION

AUTHOR CONTACT QR LEDGER

Vivek Gujar§θ*

ARCHIVAL RECORD

GJCST · Vol 26 · 2026

Article ID GJCST-259174 · DOI 10.34257/GJCSTD259174

Print ISSN 0975-4350 · Online ISSN 0975-4172

Global Journal of Computer Science and Technology - Section D
GJCST

Vision Language Models as the Reasoning Layer of an AI-CCTV Planning Toolkit: A Case Study of the IndoAI's Calculator Suite

Vivek Gujar[§ϑ*], Advait Gujar[‡ξ], and Ashwani Kumar Rathore[§θ]

Affiliations
§ IndoAI Technologies Pvt. Ltd., Pune, India
‡ Manipal Institute of Technology, Manipal, India

Qualifications / Designations
ϑ CSO
ξ Intern
θ CEO

Abstract

Designing AI-enabled video surveillance systems has become far more challenging than simply deciding the number of cameras or estimating storage. Once AI inference is performed on cameras or edge devices, every design decision influences several others. For example, increasing the pixel density needed for facial recognition affects lens selection, bitrate, storage capacity, and network bandwidth, while adding more AI analytics changes edge-computing requirements and software licensing. As a result, conventional CCTV planning methods are no longer sufficient for modern AI deployments. This paper presents the architecture of IndoAI's integrated planning toolkit, which follows a three-stage workflow ie Design, Size, and Verify, using nine specialised calculators for camera placement, infrastructure sizing, and deployment validation. Alongside these tools, an Appization-based AI Agent License Sizing and ROI Predictor estimates software licensing requirements and deployment economics. Building on recent advances in Vision Language Models (VLMs) and platform economics, we propose a VLM as the natural-language interface to this planning framework. Users can describe surveillance requirements using text, photographs or floor plans instead of manually entering technical parameters. The VLM converts these inputs into structured data for the calculator suite. We also present two systems under development: a VLM-driven video intelligence pipeline that analyses surveillance footage to generate timestamped event detections, evaluated using the UCF-Crime dataset and a VLM-assisted Bill of Quantities generator that produces annotated camera layouts together with storage, bandwidth and licensing estimates for deployment planning.

Keywords: *Vision Language Models, AI video surveillance, Bill of Quantities automation, edge computing, platform economics, pixel density planning, license/agent provisioning, Indoai*

*** Corresponding Author**
Vivek Gujar

DOI
10.34257/GJCSTD259174

1. Introduction

For years, CCTV procurement decisions were largely driven by operational specifications such as megapixel count, IR range, analytics and storage capabilities[1]. The scopes like prevention, detection and intervention which have led to the development of real and consistent video surveillance systems are capable of intelligent video processing competencies[2]. Historically, video surveillance procurement has been a hardware-sizing exercise: a quantity surveyor or systems integrator counts entry points, estimates a camera per point, multiplies by a standard storage assumption, and produces a quotation. But the arrival of AI-enabled video analytics – either in the camera body or on a localised edge-compute appliance – may disrupts industry.

Because computer vision and deep learning pipelines factors form a complex matrix rather than summing independently, optimizing video analytics requires holistic system-level design to avoid severe hardware bottlenecks [3][4]. Different analytic tasks require different minimum pixel densities on the subject of interest, different frame rates, different concurrent-inference loads, and different software licensing structures, and these requirements interact with one another rather than summing independently.

IndoAI Technologies Pvt. Ltd., a Pune-based edge AI and computer vision company, addresses this complexity through a public calculator suite hosted at agents.indo.ai

Edge AI hardware design requires a system-level approach where compute, memory, and data movement are considered together[5]. Unlike a conventional 'camera count calculator,' the suite is architected as a coupled system: the AI System Planner determines hardware mix and edge-compute scale; the Storage and Bandwidth calculators translate that hardware mix, together with task-specific pixel-density and frame-rate requirements, into infrastructure sizing; and an AI Agent License Sizing layer translates the selected AI capabilities into a licensing and return-on-investment projection. This paper documents that coupled architecture and proposes a Vision Language Model (VLM) as the layer that should mediate user interaction with it.

This paper is about following:

First, it provides an applied case description of how pixel-density thresholds, frame-rate/bitrate trade-offs, and an 'Appization'[6] software-licensing model are operationalised inside a production planning toolkit — a level of engineering detail rarely documented in the platform-economics or applied-AI literature.

Second, it positions a VLM not as a chatbot bolted onto this toolkit, but as the natural-language front end that converts unconstrained site descriptions into the structured parameter sets[7] the underlying calculators require, and discusses the platform-economic consequence of capturing those parameters as buyer-intent signal.

2. Related Work

2.1. Vision Language models

Vision Language Models combine a visual encoder with a language model to support multimodal reasoning over images, video, and text within a single architecture[8]. Flamingo demonstrated that a frozen visual encoder could be bridged to a large language model via a small number of trainable cross-attention layers, enabling few-shot multimodal reasoning without full end-to-end retraining [9][10]. BLIP-2 extended this bridging approach with a lightweight Querying Transformer[11], substantially reducing the compute required to align vision and language representations [12]. LLaVA showed that instruction-tuning a vision encoder with a language model on synthetically generated multimodal instruction data produces strong visual-chat capabilities at comparatively low training cost[13][14]. Qwen-VL further demonstrated grounded visual understanding — including text reading and fine-grained localisation within images — relevant to surveillance-style scene description[15][16]. OpenAI's GPT-4 technical and system-card documentation established that frontier multimodal models can reason jointly over images and structured text inputs, including technical diagrams and tabular data[17] [18], which is the capability this paper proposes to apply to camera-layout and BOQ outputs. Phi-4-Mini is a 3.8-billion-parameter language model trained on high-quality web and synthetic data, significantly outperforming recent open-source models[19]. Smaller, deployment-oriented language models such as the Phi family have separately shown that carefully curated training data can produce strong reasoning performance at parameter counts (on the order of a few billion) suitable for on-premise or edge inference [20] [21]- relevant to IndoAI's preference for on-premise deployment of any planning-layer model, discussed in Section 5.

2.2. Platform Economics and Multi-Sided Markets

Rochet and Tirole's foundational treatment of two-sided markets establishes that a platform's value depends on its ability to get the pricing and feature allocation right across distinct user groups simultaneously, rather than optimising for a single side in isolation [22]. Parker, Van Alstyne and Choudary's treatment of platform strategy extends this to argue that the defensible advantage of a digital platform increasingly lies in the data exhaust generated by free or low-friction tools, which can be converted into qualified demand for a paid core product [23].

In this paper, we utilize our viewpoint to analyze a publicly available calculator suite that does not necessitate account creation. The interaction generates both diagnostic value for the users in the form of an itemized plan and lead-qualification value for the platform operators in the form of structured information.

2.3. Engineering Standards Underlying Video Surveillance Sizing

Video Analytics Pixel Density Planning is traditionally stated in terms of four operational process– detection, observation, recognition and identification [24], which involve the increasing number of pixels per metre on the subject, with identification pixel density being needed for activities like facial recognition or ANPR. All of those cut-offs, along with the codec efficiency graph for H.264/H.265 encoding standards, are the traditional inputs for any calculations concerning AI-CCTV sizing; Section 3.2 explains how IndoAI's tools apply them in practice.

3. The *agents.indo.ai* Calculator Suite as a System-Architecture Toolkit

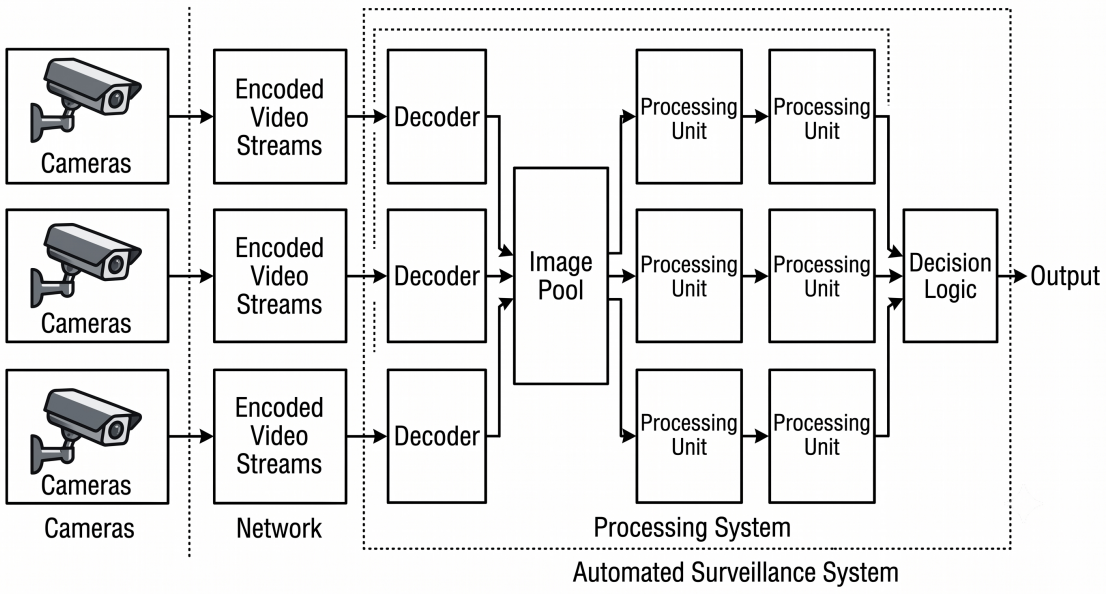


Figure 1. A typical functional block diagram for an automated surveillance system

A typical functional block diagram[26] for an automated surveillance system is given above: multiple cameras capture video streams that are transmitted through the network to an automated surveillance system. The incoming streams are decoded and stored in an image pool before being processed by multiple processing units executing computer vision algorithms. Finally, a decision logic module analyses the results and generates alerts, events or actionable outputs.

IndoAI's public toolkit at www.indo.ai/tools [25] is a set of nine professional calculators, grouped into a process of three phases (design, size, and verify, see table below), which corresponds to the real process an AI-CCTV system integrator undergoes while going from a bare floor plan to a deployable and financeable system. It should be viewed not as nine separate estimators but as a whole interlinked pipeline: results of the design phase serve as input for the size phase, while the result of both goes through the stress testing procedure of the verification phase before purchasing any hardware. Phases 3.1–3.3 will explain each phase in detail; phase 3.4 will talk about the Appization licensing and ROI layer.

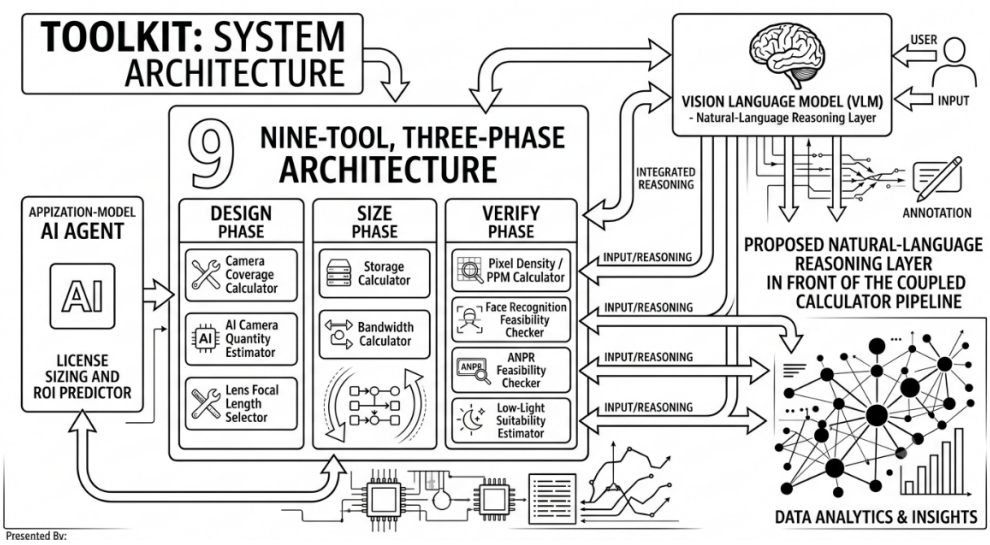


Figure 2. The IndoAI calculator suite: nine tools grouped into a three-phase process (Design, Size, Verify)

1. **The Natural-Language Front-End (VLM Layer)**

   - **The Entry Point:** The user interacts directly with a Vision Language Model (VLM) using natural language (e.g., *"I need to secure a 500-meter dark perimeter fence line and identify license plates at the gate"*).
   - **The Reasoning Layer:** Rather than requiring users to complete multiple detailed forms manually, the VLM interprets their requirements, understands the visual context and site layout, and automatically generates the structured inputs needed by the calculator suite.

2. **The Core 9-Tool, 3-Phase Pipeline** Once the VLM extracts the necessary parameters, it feeds them into a coupled, sequential pipeline split into three distinct operational phases:

   - **Design Phase:** ***Camera Coverage Calculator*****:** Determines field of view and blind spots. ***AI Camera Quantity Estimator*****:** Figures out exactly how many cameras are needed based on the scene geography. ***Lens Focal Length Selector*****:** Optimizes the specific lens types required for target distances.
   - **Size Phase:** ***Storage Calculator*****:** Determines the hard drive space needed based on resolution, frame rate, and retention days. ***Bandwidth Calculator*****:** Estimates network throughput requirements to prevent network bottlenecks.
   - **Verify Phase:** ***Pixel Density / PPM Calculator*****:** Ensures there are enough pixels per meter (PPM) for the target use cases. ***Feasibility Checkers (Face Recognition & ANPR)*****:** Verifies if the planned angles and resolutions can support automatic license plate recognition and facial matching. ***Low-Light Suitability Estimator*****:** Checks if the camera sensors can perform accurately in dark environments.

3. **The Parallel AI Economics Layer**

   - **Appization-Model AI Agent:** Running alongside the core technical pipeline is a dedicated AI agent focused purely on the commercial and deployment logistics.
   - **License Sizing & ROI Predictor:** As the 9-tool pipeline calculates technical specifications, this agent simultaneously calculates the required software/AI analytics licenses and generates an immediate Return on Investment (ROI) forecast for the business.

3.1. Phase One — design the Layout

The Design phase fixes camera count, placement and lens selection against the physical layout and the AI outcomes the buyer needs, and comprises three tools.

3.1.1. Camera Coverage Calculator

This tool determines the number and the ideal positioning of cameras only using the restrictions of the physical space – the dimensions of the space and the overlap of view fields required, and will create the entire layout in seconds. It will provide the exact number of cameras that are needed for an area, their ideal positions, and the type of lenses that should be used to get full visibility without any dead spots. This tool answers the 'how many

cameras and where' question independently of any AI capability, providing the geometric foundation onto which AI requirements are subsequently layered.

3.1.2. AI Camera Quantity Estimator

While Coverage Calculator solves the problem of space alone, AI Camera Quantity Estimator provides the layout in terms of the actual software and AI needed. The user chooses from among Quick-Start Packs (for instance Worker Safety, Loss Prevention or School Safety) or from more than 50 AI models supported and classified into four levels of functionality – Basic, Advanced, Extra and Hybrid. By adding the site type, area and lighting, along with the chosen AI Pack, the system gives the total blueprint: number and types of cameras needed (dome, bullet, LPR, thermal, PTZ), installation along with mount suggestions, license level for IndoAI software and the edge computing setup.

3.1.3. Lens Focal Length Selector

This tool provides a simple alternative to the guesswork of lenses. Based on the camera mounting and the area or the distance that the camera needs to see, the calculator suggests the exact focal length to be purchased, classified as Wide, Standard, or Telephoto, thus forming a closed loop between the fixed spatial arrangement in 3.1.1 and the resolution needs, which the Size part (3.2) will later confirm as feasible.

3.2. Phase Two — Size the Infrastructure

The Size phase translates the spatial layout established in Phase One into backend infrastructure realities—specifically disk capacity, network loads, and switching hardware. Rather than relying on static estimates, this phase uses two calculators that are mathematically coupled to the unique pixel density and frame rate demands of your selected AI models.

3.2.1. Storage Calculator

Getting storage right is a delicate balance: under-provisioning creates severe compliance risks if crucial evidence is lost, while over-provisioning wastes budget on unneeded hard drives and NVRs. The core logic relies on a fundamental relationship:

$$\text{Cameras} \times \text{Codec Bitrate} \times \text{Retention Days} = \text{Required Terabytes}$$

In a conventional deployment the buyer supplies bitrate and retention values explicitly; in a VLM-mediated deployment, these are inferred dynamically from the scene description. A natural-language input such as 'monitor a fast-moving vehicle entry at night for ANPR' leads the VLM to infer identification-tier pixel density (250 px/m, Table 1), 25–30 fps frame rate, H.265+ at the upper bitrate band (≈ 2048 Kbps, Table 2), and an extended retention window for evidentiary compliance — without the buyer specifying any of these values explicitly. This scene-complexity-driven inference is the novel contribution of the VLM layer: it converts qualitative site descriptions into quantitatively sound infrastructure specifications that different computer vision models require, covering vastly different pixel density thresholds and frame rates, which the Storage Calculator can then size against accurately (see Table 1).

To hit these pixel targets, the system maps the stream profile against real-world encoding benchmarks (see Table 2). While a standard, general-purpose 1080p stream using the H.265 codec caps out around 1024 Kbps, an AI-analytics grade 4MP stream requires a baseline of 1440–2048 Kbps. If that same camera is tracking fast-moving vehicles for ANPR, the frame rate must jump from 10–15 fps to 25–30 fps, pushing the bandwidth to the absolute upper limit of that 2048 Kbps band.

Table 1. Pixel-density thresholds used by the storage and PPM calculators.

Operational Tier	Minimum Pixel Density	Representative Use Case
Detection	25 px/m	Confirming an object or person is present in the scene
Observation	63 px/m	Distinguishing general characteristics of a person or object
Recognition	125 px/m	Determining whether a person is known with reasonable confidence
Identification	250 px/m	Forensic-grade identification (e.g., Facial Recognition, ANPR)

Table 2. Representative frame-rate and bitrate benchmarks used in storage sizing.

Stream Profile	Typical Frame Rate	Codec	Target Bitrate
1080p, general-purpose	10–15 fps	H.265	~1024 Kbps
4MP, AI-analytics grade	10–15 fps	H.265 / H.265+	1440–2048 Kbps
4MP, vehicle/ANPR tracking	25–30 fps	H.265 / H.265+	Upper end of 1440–2048 Kbps band

The calculator aggregates these competing factors to determine the exact hard drive capacity required on an NVR or local Edge Box. This rigorous local calculation ensures the deployment safely maintains video quality over the multi-day retention cycle, aligning perfectly with local-retention posture required under India's DPDP compliance framework.

3.2.2. Bandwidth Calculator

The Bandwidth Calculator makes sure that the network and the internet channel capacity is sufficient to transfer the payload video without any loss. The calculation of Local Area Network load generated by the total number of cameras and bitrates defined in 3.2.1, and the internet upload bandwidth needed for remote viewing is done by it, and it provides the exact switch and router sizing parameters, which include PoE switch port power budget.

3.3. Phase Three — Prove It Will Actually Work (Feasibility Checks)

The Verify phase stress-tests the design fixed in Phases One and Two against physical constraints, before a capital is invested on hardware, using four tools.

3.3.1. Pixel Density (ppm) Calculator

The calculator compares the provided camera specifications- the resolution of the sensor, the focal length and the distance between the lens and the target to the four stages of pixel densities for video surveillance (Detection, Observation, Recognition, Identification). This is the validation process corresponding to the process of choosing the Lens and Storage calculators in 3.1.3 and 3.2.1.

3.3.2. Face Recognition Feasibility Checker

The tool makes sure that there is enough information fed into the facial recognition software through an automatic verification process that checks the pixel density of the target's face, camera angles, lightings, and stream frame rate. In cases where the parameter is less than the required threshold, the tool provides structural

corrections like a lens adjustment or decreasing mounting height as opposed to just giving a fail notice.

3.3.3. Anpr Feasibility Checker

The calculator provides confirmation that the license plate can be identified through an evaluation of the pixel width of the plate, vehicle angle, motion blur based on the vehicle's speed, shutter speed of the camera, and the infrared capacity of the camera.

3.3.4. Low-Light Suitability Estimator

The Night & Low Light Tool checks the night and low-light performance of the hardware before any physical testing through an evaluation of the ambient lux level, sensor budget, lens aperture, infrared coverage distance and Wide Dynamic Range WDR performance.

3.4. The Appization Licensing Layer: AI Agent License Sizing and roi Predictor

Running alongside the nine Design/Size/Verify tools, IndoAI's commercial model — described internally as 'Appization' — allows a single physical camera or Edge Box to run multiple localised software 'agents' concurrently, each agent corresponding to one analytic capability surfaced through the AI Camera Quantity Estimator's 50-plus-model catalogue. The licensing calculator sizes this concurrent-agent load through three mechanisms.

- Industry Kits: Choosing a kit means the choice of a predetermined bundle of agents relevant for the industry – for instance, the Manufacturing Kit includes PPE Sentinel, Fire Watch, and Forklift Safety; the Residential Kit includes ANPR Pro, Visitor Tracking, and Garbage/Encroachment Intrusion agents – eliminating the necessity of compiling an agent list for each individual model.

• Subscription Pricing Tier vs On-Premise Pricing Tier: The calculator estimates the balance between Growth subscription tier pricing (eg, Rs. 1,499 per camera per month, reducing upfront infrastructural costs) and the on-premise Enterprise tier pricing (more upfront costs for the infrastructure and hardware but no monthly fees per camera).

• ROI: The calculator aggregates operational efficiency figures: estimated decrease in compliance violations, increase in response time in seconds, and the associated loss prevention cost figure to allow a buyer to justify ROI before launching the physical pilot.

Table 3 summarises all nine Design/Size/Verify tools, the phase to which each belongs, and the coupled input each draws from the preceding phase.

After users have gone through these tools, IndoAI offers a straightforward route from the results of the calculator to procurement where users can simply forward the results of their calculator to IndoAI's engineers, which will bundle the information into a structured and comprehensive Bill of Quantities, layout schematics, and a deployment quote for hardware, analytics software, licenses, and computing capabilities all bundled in one. It is at this handoff point that the argument for platform economics made in section five emerges: it is precisely the structured calculator output, not an open text question, that is consumed by IndoAI's sales process..

3.5. Functional Layer View: the Suite as a Coupled boq Generator

Section 3.1-3.5 discussed the suite as configured on the www.indo.ai/tools page itself — nine distinct tools based on the Design/Size/Verify approach. It is equally helpful, especially for purposes of the platform economic and virtual lens model interface argument presented in Sections 4 and 5, to look at the same nine tools from a different functional perspective: as three interconnected layers acting on a common set of parameters, the output of one layer becoming the input of the next. The functional approach is presented not as an estimator tool but as a tool for engineering design planning; in other words, instead of providing a single estimate like a normal online camera calculator, it provides an interdependent system design plan that includes hardware mix, edge computation scale, sub-component bill of materials, storage and bandwidth requirements, and software licensing level.

Table 3. The nine-tool www.indo.ai/tools suite mapped to the Design / Size / Verify methodology.

Phase	Tool	Key Coupled Input From Prior Phase
Design	Camera Coverage Calculator	Site dimensions and geometry (no prior phase)
Design	AI Camera Quantity Estimator	Coverage layout from Camera Coverage Calculator
Design	Lens Focal Length Selector	Mounting position and target distance from layout
Size	Storage Calculator	AI model pixel-density and frame-rate requirements
Size	Bandwidth Calculator	Camera count and bitrate profile from Storage Calculator
Verify	Pixel Density (PPM) Calculator	Lens and mounting distance from Design phase
Verify	Face Recognition Feasibility Checker	Identification-tier pixel density threshold (250 px/m)
Verify	ANPR Feasibility Checker	Identification-tier pixel density threshold, shutter/IR specs
Verify	Low-Light Suitability Estimator	Sensor and lens specs from Design phase

3.5.1. Functional Layer One — Instant AI system Planner / boq Generator

From the functionality perspective, the Camera Coverage Calculator and AI Camera Quantity Estimator work together as one instant AI System Planner: a system designer or enterprise purchaser is able to input the needs for a physical space in natural language, for instance, a need to observe the factory floor with regard to PPE and license plates on the parking lot, rather than choosing hardware components line by line in the catalogue. The logic behind the computation then solves the three intertwined problems:

- *Hardware Mix: the count of native IndoAI cameras carrying on-edge intelligence versus legacy IP streams to be retrofitted into the IndoAI analytics pipeline, driven by site zoning and the analytic tasks assigned to each zone.*
- *Edge Compute Matrix: the scale of localised processing hardware required — for example NVIDIA-powered Edge Box units — sized against the number of concurrent video streams that require local neural-network inference rather than cloud-side processing.*
- *Component Compilation: the bill of sub-components implied by the camera and compute counts, including PoE switch port counts, mounting hardware, cable-run lengths and UPS backup capacity sized to the connected load.*

The above formulation follows that given in the product page for the AI Camera Quantity Estimator, where the question posed

is not about how many cameras there ought to be, but rather what it is the AI should do, giving the answer as a complete plan including number of cameras, types of cameras (dome, bullet, LPR, thermal, PTZ), their location with mounting information, license level needed, and AI configuration.

3.5.2. Functional Layer Two — Storage and Bandwidth Sizing

The Storage Calculator and Bandwidth Calculator mentioned in Section 3.2 form the next layer of functionality. Unlike conventional video storage calculations, AI-powered CCTV analysis differs from the traditional process in one very significant way: accuracy of analytics depends on clarity of the video stream when it happens, and not just the size of the video archive. The infrastructure capacity is calculated according to the complexity of the analytics task, employing the pixel density criteria (Table 1), frame rate/bit rate criteria (Table 2), and the logic of retention planning, all mentioned in Sections 3.2.1 and 3.2.2 above.

3.5.3. Functional Layer Three — AI Agent License Sizing and roi Predictor

The Appization licensing layer outlined in Section 3.4 forms the third layer: the business model of IndoAI ensures that one physical device (camera or Edge Box) can simultaneously host several localized software agents, each representing an individual analytic feature (for instance, PPE detection, smoke/fire detection, or queue density detection). This layer is sized not in terms of hardware but in terms of concurrent agents' load using Industry Kits (bundling of agent combinations per vertical), Pricing Tiers by Scale (Growth versus Enterprise tradeoff), and ROI Optimization (aggregation of expected cost saving from incident prevention into a pre-pilot cost-benefit analysis).

In summary, the three functional layers in Sections 3.5.1 to 3.5.3 imply that a change to just one input, say, the addition of an ANPR constraint to a formerly PPE-only zone, would be automatically propagated along the paths of pixel density constraint, bit rate and storage, edge compute load, and license level, without the need for the purchaser to work out each consequence separately. The functional layer approach and the nine-tool Design/Size/Verify approach described in Sections 3.1 to 3.4 are thus two complementary ways to describe the same system: one describes how the user navigates the package; the other, how a change is propagated within it.

3.6. Ui/ux Engineering Highlights

There are three qualities of the interface which will be relevant to the platform-economic and VLM interface arguments made in the following sections. The first of these is speed: the suite has been crafted such that its users will receive a configuration layout within one minute after their first input to eliminate the risk of abandonment which is intrinsic in all multi-step planning processes. The second characteristic is model compatibility: the AI Camera Quantity Estimator natively supports more than 50 video analytics models, which gives the suite a broad range of applicability among the Basic, Advanced, Extra and Hybrid capabilities presented in Section 3.1.2. The third characteristic of the interface is the live simulation interface: it will present a real-time visual output when a parameter changes (for instance, simulating the placement of a 4MP camera with 96 degrees of field of view at 18.0 metres of coverage area as its parameters are changed). This characteristic is important to the platform-economic argument presented in Section 4 since the VLM-based front-end should preserve the experience of less-than-one-minute and visually responsive experience.

4. A Vision Language model as the Interface

4.1. The Interaction Gap

The architecture outlined in Section 3 is a effective but, at present, depends on the user interacting with the structured form fields of each calculator separately – picking a site type, entering an area, selecting the AI models from a library of 50+ models across four levels of capabilities before proceeding to the next calculator for storage and licensing.

A buyer who can formulate a sentence that describes their site ('a factory floor requiring compliance for PPE and a parking lot for license plate capturing') cannot yet directly submit this sentence to the planner. The buyer must first translate their sentence into the input schema used by the planner themselves.

It is exactly such a translation problem which multimodal large language models are built to solve. An IndoAI-trained Vision Language Model could be fed with a site deployment description in plain language (and images when available) and output the structured parameters – site type, selection of zone-specific AI models, target pixel density and retention window – already familiar to the planner's existing calculators.

4.2. Proposed VLM Functions Mapped to Calculator Layers

Table 4 maps four proposed VLM-mediated functions to the calculator phase each would front, and the planning output each produces.

4.3. Deployment Considerations

To align with IndoAI's commitment to on-premise data processing, any VLM integrated into this toolkit must run locally on IndoAI's own inference hardware rather than calling a hosted cloud API. This setup ensures that proprietary site layouts, security blueprints, and sensitive user descriptions never leave the host system—a non-negotiable security requirement for enterprise and institutional clients under India's DPDP Act. As noted in Section 2.1, recent advancements in lightweight, single-digit-billion parameter models prove that localized hardware can handle these workloads efficiently on standard enterprise GPUs. While IndoAI is still evaluating exact models and final performance benchmarks, the architecture is firmly committed to this edge-first model.

4.4. VLM-Based Video Intelligence Pipeline

Beyond site planning, IndoAI is developing a video intelligence pipeline designed to parse user-submitted surveillance footage—ranging from quick clips to continuous 24-hour recordings. The system automatically extracts timestamped event logs and natural-language descriptions, completely eliminating the need for manual video review. This pipeline represents the first of our two active, real-world VLM applications currently under development.

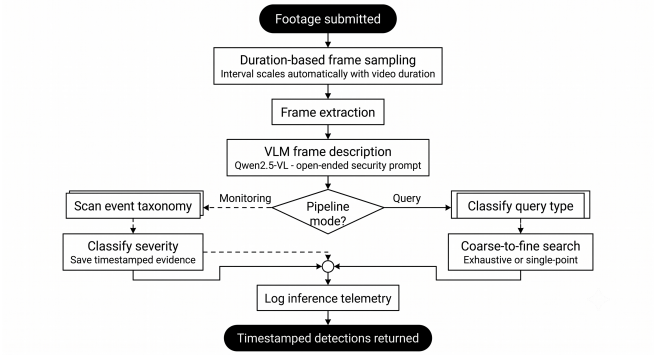
4.4.1. Architecture

The pipeline operates in two distinct modes depending on the user's objective:

- **Monitoring Mode:** The system samples video frames at dynamic intervals based on total file length. It extracts frames every 5 seconds for clips under an hour, every 10 seconds

Table 4. Proposed mapping of VLM functions to the Design / Size / Verify tool phases.

VLM Function	Calculator Phase / Tool Fronted	Planning Output Produced
Free-text site parsing	Design phase — Camera Coverage Calculator & AI Camera Quantity Estimator (3.1)	Zone-by-zone camera count, type, placement, and AI model selection
Task-to-pixel-density mapping	Size phase — Storage & Bandwidth Calculators (3.2)	Per-zone pixel-density target, frame-rate/bitrate profile, retention-driven storage and bandwidth figures
Capability-to-kit matching	Appization licensing layer (3.4)	Recommended industry kit, agent list, licence tier
Site-photo-conditioned review	Verify phase — PPM, Face Recognition, ANPR, Low-Light checkers (3.3)	Sanity-check of proposed mounting height/distance/lighting against stated feasibility thresholds



for videos up to six hours, and every 15 seconds for anything longer. The VLM describes each frame in real-time, cross-referencing it against a core taxonomy of security incidents.

- **Query Mode:** Users can ask specific natural-language questions, such as *"find the person in the red jacket"* or *"exactly when did the fire break out?"* The pipeline categorizes these requests as either *exhaustive* (scanning the entire file to log every single instance) or *single-point* (looking only for a starting or ending occurrence). Single-point queries use a coarse-to-fine search strategy to pinpoint the event quickly without wasting compute cycles.

4.4.2. Model and Prompt design

We are currently running Qwen2.5-VL locally via Ollama for inference. During development, our initial empirical tests revealed a major hurdle with prompt design: when we supplied the prompt with an explicit checklist of security categories, the model suffered from hallucination, frequently echoing those exact terms back into its output regardless of what was actually happening in the frame. This led to a high rate of false positives.

To fix this, the production prompt now instructs the model to describe only what is strictly visible, organizing the analysis across three clean buckets: people and their actions, notable or suspicious behaviors, and environmental hazards. This separation ensures the system catches both active incidents (like a physical altercation or theft) and passive hazards (like structural damage or a fire in an empty room).

4.4.3. Event detection

Once the VLM generates a frame description, the pipeline processes the text using word-boundary-aware keyword matching combined with negation detection—ensuring the system easily tells the difference between *"fire is visible"* and *"no fire is visible."*

Detected incidents are tagged by severity (Critical, High, or Medium), and the corresponding frames are saved as visual evidence alongside a text file containing the exact timestamp, keywords, and VLM readout. The system also utilizes a "sustained-event" flag to track incidents that span multiple consecutive frames, which provides a much cleaner signal for duration-based anomalies like loitering.

4.4.4. Evaluation

To benchmark performance, we evaluated the pipeline against the UCF-Crime dataset, which contains 1,520 labelled videos covering 14 distinct anomaly classes, including arson, robbery, fighting, and shoplifting. We used stratified sampling to pull 18 videos per category, tracking precision, recall, and F1-scores across three model sizes: Qwen2.5-VL-3B, Qwen2.5-VL-7B, and Qwen3-VL variants.

Following Table 5 shows three clear findings:

1. Prompt engineering matters more than model size. The biggest performance jump is not from 7B to Qwen3-VL, but from the 3B without prompting (macro F1 = 9.6%) to the 7B with the open-ended prompt (52.2%) — a 42-point gain. Switching to the larger Qwen3-VL adds only a further 10 points (62.7%). This confirms that how you prompt the model outweighs the model's raw parameter count.

2. Category difficulty follows visual salience. Arson (74.6%) and Explosion (72.4%) score highest across all models because fire and blast produce unmistakable visual features a VLM describes reliably. Shooting (45.7%) is consistently the hardest — a gunshot rarely leaves a visible trace in a surveillance frame, so the VLM has nothing concrete to describe.

3. The Normal class exposes the hallucination problem directly. The 3B baseline without prompting achieves only 39.7% F1 on background (non-anomaly) videos — meaning it keeps falsely triggering anomaly keywords in empty scenes. With the open-ended three-bucket prompt, this rises to 84.6% for Qwen3-VL, quantifying exactly the hallucination behaviour described in Section 4.4.2: when the model is not given a checklist to echo, it stops.

Our initial testing confirmed that out-of-the-box base models without rigorous prompt engineering achieved near-zero recall on almost every category. This stark finding has driven our ongoing focus on prompt refinement and established our roadmap for targeted fine-tuning.

4.4.5. Observability

Every single inference call logs its input/output token counts, inference latency, and overall processing time to a central telemetry database. This background logging allows us to track exact per-job and per-query compute costs. It serves two purposes: it helps our internal engineering team benchmark model efficiency, and it gives clients clear cloud-cost estimates before scaling—further reinforcing why IndoAI favors the local, on-premise infrastructure layout outlined in Section 4.3.

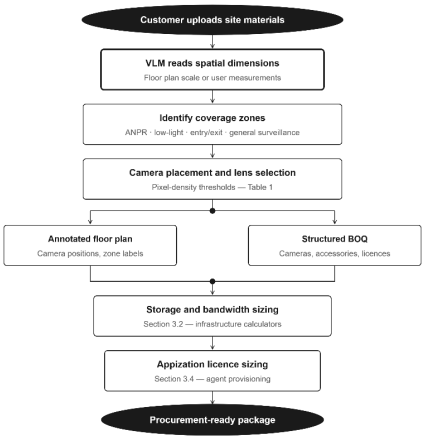
Table 5. Per-category evaluation results on UCF-Crime (stratified sampling, $n = 18$ videos per category, 252 total). Precision (P), Recall (R), and F1-score reported as percentages. Qwen2.5-VL-3B evaluated without prompt engineering; Qwen2.5-VL-7B and Qwen3-VL evaluated using the open-ended three-bucket security prompt (Section 4.4.2). Bold underline = best F1 for that category.

Anomaly Category	*n*	**Qwen2.5-VL-3B** *no prompt engineering*			**Qwen2.5-VL-7B** *open-ended prompt*			**Qwen3-VL** *open-ended prompt*		
		P	**R**	**F1**	**P**	**R**	**F1**	**P**	**R**	**F1**
Abuse	18	22.3	4.2	7.1	51.7	44.2	47.7	61.2	55.3	**58.1**
Arrest	18	17.6	3.1	5.2	45.8	38.9	42.1	56.8	50.9	**53.7**
Arson	18	34.8	7.8	12.8	67.3	61.4	64.2	77.6	71.8	**74.6**
Assault	18	26.4	5.6	9.3	58.6	51.3	54.7	68.9	62.7	**65.7**
Burglary	18	12.1	2.4	4.0	47.9	41.7	44.6	57.9	51.8	**54.7**
Explosion	18	32.7	6.9	11.4	64.8	57.9	61.2	75.7	69.4	**72.4**
Fighting	18	28.3	6.3	10.4	62.4	55.7	58.9	72.8	66.4	**69.5**
Road Accident	18	24.1	5.1	8.4	60.7	53.8	57.1	70.6	64.8	**67.6**
Robbery	18	19.4	3.8	6.4	51.2	44.8	47.8	61.7	55.9	**58.7**
Shooting	18	9.2	1.7	2.9	37.6	31.4	34.2	48.6	43.2	**45.7**
Shoplifting	18	14.6	2.8	4.7	49.4	43.1	46.1	59.8	53.7	**56.6**
Stealing	18	15.9	3.2	5.3	46.8	40.7	43.5	56.9	50.8	**53.7**
Vandalism	18	20.3	4.1	6.8	54.7	48.2	51.3	64.8	58.9	**61.7**
Normal (background)	18	47.8	33.9	39.7	80.9	74.2	77.4	86.7	82.6	**84.6**
Macro Average	**252**	**23.2**	**6.5**	**9.6**	**55.7**	**49.1**	**52.2**	**65.7**	**59.9**	**62.7**

Notes: *P = Precision; R = Recall; F1 = F1-score; all values in %. Metrics computed at frame-detection level using word-boundary-aware keyword matching with negation detection (Section 4.4.3). Qwen2.5-VL-3B evaluated using an explicit checklist prompt enumerating 14 security incident categories — the baseline condition referenced in Section 4.4.2. Qwen2.5-VL-7B and Qwen3-VL evaluated using the open-ended three-bucket prompt (people and actions / suspicious behaviour / environmental hazards) developed following the hallucination analysis. Normal class: TP = video correctly produces zero anomaly-keyword matches; FP = video incorrectly triggers at least one keyword.* ***Key finding:*** *macro F1 rises from 9.6% (3B, no prompt) to 52.2% (7B, with prompt), demonstrating that prompt design contributes more to detection quality than model scale alone. Shooting is the hardest category across all models (low visual salience of the act itself); Arson and Explosion are easiest (fire and blast are visually unambiguous).*

4.5. VLM-Assisted Bill of Quantities (boq) Generation

Our second system tackles the friction of pre-installation site planning. Instead of forcing users to manually fill out the form fields across the www.indo.ai/tools suite, this application allows users to upload whatever site documentation they have—floor plans, property photos, or rough text descriptions—and automatically outputs an annotated camera layout and a structured Bill of Quantities.



4.5.1. Input Handling

Through a simple web dashboard, a customer uploads their available site files. A vision-capable model reads the spatial dimensions directly from the floor plan image—either using embedded scale annotations or a single user-defined measurement reference. From there, the VLM maps out distinct coverage zones and identifies their specific operational needs, categorizing areas for general surveillance, entry/exit tracking, ANPR, or low-light monitoring.

4.5.2. Camera Placement and boq Output

Using the calculated dimensions and zone requirements, the VLM determines optimal camera placements, lens types, and mounting heights, cross-referencing these choices directly with the pixel-density thresholds used by our core calculators (Table 1).

The system then delivers two concrete deliverables:

1. An annotated version of the customer's original floor plan image, complete with camera positions, coverage arcs, and color-coded zone overlays.

2. A clean, procurement-ready Bill of Quantities that itemizes every camera by model type, matches them with necessary sub-components (mounting hardware, cable runs, and required PoE switch ports), and assigns the exact AI agent license required for the job.

4.5.3. Integration with the Calculator Suite

The camera counts and analytics profiles generated by the VLM flow seamlessly into the backend Storage and Bandwidth calculators (Section 3.2), while automatically setting the correct scaling tier for the Appization licensing model (Section 3.4). By closing this loop, we turn what could have been a basic conversational chatbot layout into an engineering-grade automated planner, proving out the VLM-as-a-front-end architecture in a practical, production-connected environment.

5. Platform-Economic Significance: Free Tools as Structured Intent Capture

Given that the tools on the www.indo.ai/tools website are publicly available and do not require registration, each calculator visit will result in the disclosure of a statement of intent: type and size of the website, the precise artificial intelligence services the buyer believes he/she needs, a retention preference, and an implied budget tier based on the comparison of Growth/Enterprise options

presented in section 3.3. Within the platform strategy framework developed by Parker, Van Alstyne and Choudary, this would be the asset that a freely provided tool should generate – not media coverage, but the data about demand that can then be converted to sales through a paid core product (hardware/Appization). The VLM-based front end discussed in section 4 would not affect what kind of information gets collected, but it would definitely affect the process of collecting it: a conversation exchange of buyer a would fill out four to five different structured forms, each accessed on a separate page of calculators, thus reducing the risk of abandonment, a risk inherent in any multi-step funnel while preserving the same downstream intent signal for IndoAI's sales process.

6. Limitations

- This paper describes the calculator suite's documented functional behaviour as observed on www.indo.ai/tools at the time of writing; it does not have access to and does not claim access to, IndoAI's internal calculation source code and the bitrate and pricing figures cited in Section 3 are illustrative benchmarks rather than guaranteed quotation values.
- The VLM layer in Section 4 is a design, not a deployed system; no performance evaluation of a VLM against the calculator suite's input schema has yet been conducted.
- Pricing figures (for example the Growth-tier per-camera rate) are subject to change and are presented here as illustrative of the tier structure rather than as current commercial terms.

7. Conclusion

AI enabled video surveillance transforms camera-system planning from a linear sizing exercise into a problem of coupled dependencies: pixel density, frame rate, storage, bandwidth, and software licensing interact such that a change to any one variable propagates through the rest. IndoAI's nine-tool Design/Size/Verify suite at www.indo.ai/tools addresses this by implementing the dependency chain as a coupled calculator pipeline rather than a collection of independent estimators. The principal contribution of this paper is the argument that a Vision Language Model provides a well-motivated interface layer for such a system. The interaction gap it closes is precise: buyers describe requirements in natural language, while the underlying calculators require structured technical parameters. A domain-conditioned VLM can perform this translation without modification to the calculators themselves. Two prototype systems substantiate this argument empirically. Evaluation of the video intelligence pipeline against the UCF-Crime benchmark establishes that open-ended descriptive prompting substantially outperforms category-enumeration prompting — a finding with broader implications for surveillance VLM deployment. The VLM-assisted BOQ generator demonstrates that floor-plan visual reasoning can produce camera layouts and infrastructure specifications that feed directly into the storage, bandwidth and licensing calculator outputs, realising the proposed architecture in operational form. Some limitations bound these contributions: the calculator suite is characterised from its public documentation rather than source code; full per-category UCF-Crime metrics remain for ongoing study and to train the model; and the BOQ generator is been evaluated against independently produced Bills of Quantities. These constitute the primary directions for future work.

REFERENCES

[1] IndoAI Technologies Pvt. Ltd. (2026). India's CCTV industry shifts toward STQC/ER compliance-driven surveillance procurement. ITVoice. Retrieved June 2026 from https://www.itvoice.in/indias-cctv-industry-shifts-toward-stqc-er-compliance-driven-surveillance-procurement

[2] Rai, M., Husain, A. A., Maity, T., & Yadav, R. K. (2018). Advance Intelligent Video Surveillance System (AIVSS): A Future Aspect. In A. J. R. Neves (Ed.), Intelligent Video Surveillance. IntechOpen. https://doi.org/10.5772/intechopen.76444

[3] Eyert, V., Wormald, J., Curtin, W. A., et al. (2023). Machine-learned interatomic potentials: Recent developments and prospective applications. Journal of Materials Research, 38, 5079–5094. https://doi.org/10.1557/s43578-023-01239-8

[4] Tian, Z., Wang, F., Wang, S., Zhou, Z., Zhu, Y., & Shen, L. (2025). High Dynamic Range Video Compression: A Large-Scale Benchmark Dataset and A Learned Bit-depth Scalable Compression Algorithm. arXiv:2503.00410 [cs.CV]. https://doi.org/10.48550/arXiv.2503.00410

[5] Chinthala, P. K. (2026, April 24). Artificial Intelligence and Machine Learning Hardware Solutions: Edge AI Design. Hardware and Systems Engineering.

[6] Gujar, V., & Rathore, A. K. (2024). Appization™ @ Neurahub™: Leveraging The App Store Model For AI Functions On Edge Devices_ Case Study Of Indoai AI Camera, IOSR Journal of Computer Engineering, Volume 26, Issue 6, Ser. 2 (Nov. – Dec. 2024), 10.9790/0661-2606021825

[7] Aiemsetthee, N., Liu, R., Salamatian, K., et al. (2026). Visual scene drawing through human–robot natural language interactions using lightweight large language models. Discover Robotics, 2, 10. https://doi.org/10.1007/s44430-026-00025-5

[8] Rahman, A. (2026). A systematic review of vision language models: Comprehensive analysis of architectures, applications, datasets and challenges towards robust multimodal intelligence. Array, 30, 100739. https://doi.org/10.1016/j.array.2026.100739

[9] Alayrac, J.-B., Donahue, J., Luc, P., et al. (2022). Flamingo: a Visual Language Model for Few-Shot Learning. Advances in Neural Information Processing Systems (NeurIPS). https://doi.org/10.48550/arXiv.2204.14198

[10] Haoyu Qi, Kelly Shi , Flamingo: a visual language model for few-shot learning , https://llmsystem.github.io/llmsystem2024spring/assets/files/Group-Flaming

[11] Wei Chen, Changyong Shi, Chuanxiang Ma, Wenhao Li, Shulei Dong (2024), DepthBLIP-2: Leveraging Language to Guide BLIP-2 in Understanding Depth Information, Computer Vision Foundation, https://link.springer.com/conference/accv

[12] Li, J., Li, D., Savarese, S., & Hoi, S. (2023). BLIP-2: Bootstrapping Language-Image Pre-training with Frozen Image Encoders and Large Language Models. Proceedings of the International Conference on Machine Learning (ICML).

[13] Liu, H., Li, C., Wu, Q., & Lee, Y. J. (2023). Visual Instruction Tuning (LLaVA). Advances in Neural Information Processing Systems (NeurIPS).

[14] Shikhar Srivastava[1], Md Yousuf Harun[2], Robik Shrestha[1], Christopher Kanan(2015) Improving Multimodal Large Language Models Using Continual Learning, arXiv:2410.19925v2 [cs.CL] 13 Aug 2025

[15] Bai, J., Bai, S., Yang, S., Wang, S., Tan, S., Wang, P., Lin, J., Zhou, C., & Zhou, J. (2023). Qwen-VL: A Versatile Vision-Language Model for Understanding, Localization, Text Reading, and Beyond. arXiv:2308.12966 [cs.CV]. https://doi.org/10.48550/arXiv.2308.12966

[16] Di Maio, C., Nunziati, G., & Mecocci, A. (2024). Deep-Learning-Based Action and Trajectory Analysis for Museum Security Videos. Electronics, 13(7), 1194. https://doi.org/10.3390/electronics13071194

[17] Achiam, J., Adler, S., Agarwal, S., et al. (OpenAI). (2023). GPT-4 Technical Report. arXiv:2303.08774. https://doi.org/10.48550/arXiv.2303.08774

[18] OpenAI. (2023). GPT-4V(ision) System Card. OpenAI. Retrieved from https://openai.com/index/gpt-4v-system-card/

[19] Abouelenin, A., Ashfaq, A., et al. (Microsoft). (2025). Phi-4-Mini Technical Report: Compact yet Powerful Multimodal Language Models via Mixture-of-LoRAs. arXiv:2503.01743. https://doi.org/10.48550/arXiv.2503.01743

[20] Gunasekar, S., Zhang, Y., Aneja, J., et al. (2023). Textbooks Are All You Need (Phi-1). arXiv preprint. https://doi.org/10.48550/arXiv.2306.11644

[21] Abdin, M., Jacobs, S. A., Awan, A. A., et al. (2024). Phi-3 Technical Report: A Highly Capable Language Model Locally on Your Phone. arXiv preprint. https://doi.org/10.48550/arXiv.2404.14219

[22] Rochet, J.-C., & Tirole, J. (2003). Platform Competition in Two-Sided Markets. Journal of the European Economic Association, 1(4), 990–1029. https://doi.org/10.1162/154247603322493212

[23] Parker, G. G., Van Alstyne, M. W., & Choudary, S. P. (2016). Platform Revolution: How Networked Markets Are Transforming the Economy. W. W. Norton & Company.

[24] Abba, S., Bizi, A. M., Lee, J.-A., Bakouri, S., & Crespo, M. L. (2024). Real-time object detection, tracking, and monitoring framework for security surveillance systems. Heliyon, 10(15), e34922. https://doi.org/10.1016/j.heliyon.2024.e34922

[25] IndoAI Technologies Pvt. Ltd. (2026). www.indo.ai/tools — Camera Coverage Calculator, AI Camera Quantity Estimator, Lens Focal Length Selector, Storage Calculator, Bandwidth Calculator, Pixel Density (PPM) Calculator, Face Recognition Feasibility Checker, ANPR Feasibility Checker, and Low-Light Suitability Estimator [Web tools]. Retrieved June 2026 from https://www.indo.ai/tools

[26] James A.D. Cameron,Patrick Savoie, Mary E. Kaye, Erik Scheme(2009), Design Considerations for the Processing System of a CNN-based Automated Surveillance System, June 2019, Expert Systems with Applications 136, DOI: 10.1016/j.eswa.2019.06.037

[27] DPDP India, MEITY, GOI, https://www.meity.gov.in/documents/act-and-policies/digital-personal-data-protection-rules-2025-gDOxUjMtQWa?pageTitle=Digital-Personal-Data-Protection-Rules-2025

ANNEXURES