Edge–Cloud Inference Deployment for Perception-Driven Automation

A systems evaluation of stereo-vision assisted zone-occupancy for building automation control under practical accuracy, latency, and energy constraints

Master's Thesis in Computer Science and Engineering

Alexander Däldborg, Eric Palm

Department of Computer Science and Engineering
CHALMERS UNIVERSITY OF TECHNOLOGY
UNIVERSITY OF GOTHENBURG
Gothenburg, Sweden 2026

Master's Thesis 2026

Edge–Cloud Inference Deployment for Perception-Driven Automation

A systems evaluation of stereo-vision assisted zone-occupancy for building automation control under practical accuracy, latency, and energy constraints

Alexander Däldborg, Eric Palm

UNIVERSITY OF GOTHENBURG

CHALMERS
UNIVERSITY OF TECHNOLOGY

Department of Computer Science and Engineering
Chalmers University of Technology
University of Gothenburg
Gothenburg, Sweden 2026

Edge–Cloud Inference Deployment for Perception-Driven Automation
A systems evaluation of stereo-vision assisted zone-occupancy for building automation control under practical accuracy, latency, and energy constraints
Alexander Däldborg, Eric Palm

Supervisor: Björn Praestegaard Larsen, Department of Computer Science and Engineering
Examiner: Morten Fjeld, Department of Computer Science and Engineering

Master's Thesis 2026
Department of Computer Science and Engineering
Chalmers University of Technology and University of Gothenburg
SE-412 96 Gothenburg
Telephone +46 31 772 1000

Typeset in LaTeX
Gothenburg, Sweden 2026

Alexander Däldborg, Eric Palm
Department of Computer Science and Engineering
Chalmers University of Technology and University of Gothenburg

Abstract

This thesis evaluates how edge and cloud inference affect a stereo-depth-assisted, zone-based lighting controller for building automation. The system combines person detection with depth-based spatial assignment to estimate whether predefined lighting zones are occupied. Several model configurations are compared under edge and cloud deployment using controlled scenarios and full-day analytical lighting reconstructions.

The results show that deployment placement affects both perception and system behaviour. Cloud inference improves recall in more difficult conditions, such as low light and obstruction, but introduces transport delay, queue behaviour, higher measured processing energy, and greater privacy exposure. Edge inference provides lower processing-energy use, avoids remote communication overhead, and keeps more data local, but is less robust in difficult visual conditions. Across the evaluated configurations, the projected zone-aware controller reduces lighting energy compared with less spatially specific baseline strategies.

Keywords: Edge inference, cloud inference, zone occupancy, smart lighting, building automation

Acknowledgements

Firstly, we would like to thank our supervisor, Björn Praestegaard Larsen, for his guidance, support, and constructive feedback throughout the thesis. We would also like to thank our examiner, Morten Fjeld, for his academic guidance and valuable input on the work. Special thanks to Jessica Drotz, whose support helped shape our reasoning about GDPR, consent, data collection, and the practical limitations involved in working with person-related visual data.

The thesis work was carried out with support from Knightec Group. In particular, we would like to extend our thanks to Felix Fredin, Fredrik Wulff, Andrey Alishev, Isak Holmdahl, Rasmus Standar, and Ronja Lagerqvist for their supervision, technical discussions, feedback, and practical support during the project. We are also very grateful to the participants who contributed to the data-collection activities used in the study.

Alexander Däldborg, Eric Palm, Gothenburg, May 2026

Contents

List of Figures

List of Tables

1

Introduction

Recent smart-lighting research has proposed video-based occupant positioning as a way to support more spatially aware indoor lighting control [1]. For building automation, this raises a broader systems question beyond whether occupants can be detected. A practical controller must convert detections into a zone-level control signal that remains accurate, timely, and robust during continuous operation. In that context, inference placement becomes a central architectural choice, because executing the perception model at the edge or in the cloud changes the conditions under which the control signal is produced.

The study examines this system-level problem in the context of zone-based lighting control. The system combines person detection with stereo-depth information to estimate whether predefined zones are occupied. Lighting is used as the application context because control effects are visible to occupants and because improved lighting control has been associated with energy-saving potential in prior work [2, 1]. The study is relevant to building operators, system integrators, and occupants because deployment choices affect not only perceptual performance, but also latency, resource demand, energy use, and privacy exposure [3, 4, 5].

1.1 Problem Description

Zone-based lighting control depends on more than knowing whether a room is occupied. To avoid lighting unused areas while still responding to occupants, the controller needs a spatially meaningful signal that indicates which part of the space is in use. Conventional control strategies do not always provide that level of information. Manual controls and binary occupancy-triggered controls are straightforward to commission and understand, but they provide only coarse occupancy information [6]. A manual switch can keep the tested fixture group active for longer than needed, while a binary occupancy sensor can indicate presence without determining which part of a larger space is being used. These limitations motivate their use as physical-switch and motion-detector baselines in the evaluation.

Camera-based perception offers a richer alternative because it can support more spatially aware lighting decisions. Recent work by Ning et al. demonstrates how video-based occupant positioning can support fine-grained smart-lighting control,

while also identifying practical challenges for video-based positioning under conditions such as low light, occlusion, and privacy protection [1]. Their work is primarily concerned with multi-occupant positioning, illumination-preference modelling, and lighting optimisation. It does not evaluate how alternative deployment architectures affect the perception layer that produces the information required by such control strategies.

The resulting problem is how to produce zone-occupancy information in a way that is useful for continuous lighting control. The relevant output is *zone occupancy*: whether a predefined lighting zone is occupied, not generic presence detection or full occupant-position tracking. Producing this signal requires person detections to be mapped to spatial zones, for example using stereo-depth cues, and the resulting signal must remain usable under the timing, robustness, and feasibility constraints of a practical controller.

Several architectural choices affect whether such a perception pipeline is viable. The perception task may be executed locally on an edge device or partly offloaded to the cloud, and these alternatives can change the latency, runtime burden, energy consumption, privacy exposure, and operational robustness of the system. Model configuration further affects these trade-offs, since larger or more demanding object-detection models may improve detection performance while also increasing runtime and resource demand [4].

Prior work leaves a gap between spatially aware lighting-control research and the evaluation of deployable inference systems. Existing smart-lighting studies demonstrate that richer occupant-position information can support more fine-grained control, but they do not sufficiently examine how the perception layer should be deployed under practical constraints [7, 8, 1]. Conversely, embedded-AI and offloading studies evaluate inference performance, latency, and energy, but often stop at detector- or platform-level metrics instead of tracing the consequences into a control-oriented occupancy signal [4, 9]. What remains less examined is how deployment and detector choices affect a stereo-depth-assisted zone-occupancy pipeline when accuracy, robustness, latency, runtime burden, energy use, and privacy exposure are evaluated together [5]. This study addresses that gap in the context of projected zone-based lighting control.

1.2 Purpose of the Study

The purpose of this study is to investigate how software-system architectural decisions affect the behaviour and performance of a perception-driven building-automation system. In particular, the study treats inference placement as an architectural decision: whether inference is executed locally on an edge device or offloaded to a cloud service. The study also considers model configuration as a related deployment decision, since the selected model affects not only detection quality but also latency, runtime burden, energy consumption, and operational feasibility. From a software engineering perspective, the study is concerned with how these design decisions in-

fluence system-level quality attributes. A perception-driven controller is not useful only because it can detect occupants. It must also produce a control-relevant signal that is timely, robust under difficult conditions, feasible to run continuously, energy-aware, and acceptable from a privacy perspective. The evaluation considers these decisions through their effects on the end-to-end perception-and-decision pipeline, instead of treating detector accuracy as an isolated metric.

This research purpose is operationalised through the design and evaluation of a camera-based, stereo-depth-assisted pipeline for zone-based lighting. In the pipeline, person detections are combined with stereo-depth-supported spatial assignment to estimate whether predefined lighting zones are occupied. The implemented runtime ends at zone-occupancy decisions and logging, while lighting behaviour and net-energy outcomes are evaluated through control-mode calculations instead of direct physical actuation.

More specifically, the project aims to:

- develop an end-to-end perception-and-decision pipeline for zone-based lighting using person detection and stereo-depth-supported spatial assignment,
- realise comparable edge and cloud inference variants within the same overall controller architecture,
- evaluate how inference placement and model configuration affect zone-occupancy estimation, robustness, decision-path latency, runtime burden, and operational energy consumption, and
- compare projected controller behaviour with simpler non-AI baseline strategies in controlled replay scenarios and a field-like working-day trace.

The study does not aim to introduce a new object-detection architecture or a universally applicable lighting-control method. Its contribution lies instead in a software engineering-oriented evaluation of one integrated application setting for deployable building automation: a stereo-depth-assisted, zone-based perception-and-decision pipeline in which inference placement is treated as an architectural variable for implemented edge and cloud execution. Because the system relies on camera-based sensing, privacy exposure and ethical data handling are treated as design constraints throughout the study instead of secondary concerns.

1.3 Research Questions

The study is guided by the following research questions:

RQ1
How do inference placement and model configuration affect zone-occupancy accuracy, robustness, and end-to-end decision-path latency in a stereo-depth-assisted zone-occupancy controller?

RQ2
How do inference placement and model configuration affect runtime burden and operational energy consumption during continuous perception, including communication overhead where applicable?

RQ3
Under controlled replay and field-like trace scenarios, how do the proposed controller variants affect projected lighting behaviour and projected net-energy outcomes relative to baseline automation strategies?

Together, these questions address three related but distinct concerns. RQ1 focuses on control-relevant perceptual performance. RQ2 focuses on runtime burden and energy consumption during continuous operation. RQ3 considers what those trade-offs mean at the application level when the proposed controller variants are compared with simpler automation strategies. Here, *robustness* refers to performance degradation under more challenging observed conditions, such as occlusion, illumination change, or multi-person obstruction. *Response quality* refers to the timeliness and stability of projected lighting actions, not to generic user-interface responsiveness. These questions are operationalised in Chapter 5 through explicit metrics: RQ1 through zone-occupancy accuracy, robustness-oriented scenarios, and end-to-end decision-path latency, RQ2 through logged runtime-burden and energy measures, and RQ3 through projected lighting-behaviour indicators and net-energy comparisons against baseline strategies.

1.4 Limitations and delimitations

Here, *limitations* denote factors that may affect validity or interpretation, whereas *delimitations* denote deliberate boundaries set to keep the study focused and feasible [10].

The main limitations concern empirical scope, measurement boundaries, and actuation modelling. The evaluation is based on controlled recordings, replayed SVO traces, and one field-like recording from the monitored environment. Although this material supports comparison between implemented configurations, it does not represent a long-term live deployment. The results should be interpreted with caution regarding seasonal variation, long-term maintenance, changing occupancy patterns, and model degradation over extended use.

A second limitation is that lighting actuation is not physically implemented in the prototype runtime. The implemented system produces zone-occupancy decisions and logs, while lighting behaviour and net-energy outcomes are derived from control-mode calculations. The projected lighting results indicate expected application-level consequences under the stated assumptions, not directly measured fixture-level behaviour.

A third limitation concerns measurement boundaries. Latency, timing, and power measurements depend on the implemented logging and instrumentation setup. In addition, cloud-related energy measurements include logged cloud GPU power and edge-side power, but not full cloud-host power, CPU, memory, storage, network equipment, or datacentre overhead. The reported cloud-energy results should be read as bounded operational measurements, not complete infrastructure-energy estimates.

The study is delimited to lighting as the application context, even though occupancy-aware perception may also be relevant to other building-automation functions such as HVAC or blinds. The implementation and evaluation are limited to a single-room or otherwise small-scale setting instead of whole-building deployment. The study does not introduce a new detector architecture and does not attempt a broad empirical comparison with other occupancy-sensing modalities such as NIR, CO_2, Wi-Fi, pressure, or badge-based systems. Instead, the technical scope is restricted to a camera-based, stereo-depth-assisted perception pipeline and to how alternative inference placements affect that pipeline within the same control problem.

The study does not provide a formal legal or regulatory compliance audit. Privacy, consent, and data-minimisation considerations are treated as methodological and architectural constraints, and the study is *GDPR-aware*, not a formal GDPR compliance certification.

1.5 Significance of the Study

The significance of this study lies in its combined treatment of spatially aware smart-lighting control and deployable inference-system evaluation. Prior work suggests that richer spatial sensing can support more fine-grained lighting control [7, 1], that model choice and hardware constraints affect runtime feasibility [4], and that deployment architecture influences what data is processed locally or transmitted remotely [5]. However, these concerns are not always examined together within the same control-oriented system.

This study addresses the research gap identified in Section 1.1 through a stereo-depth-assisted zone-occupancy pipeline evaluated under practical system constraints. The contribution is not a new detector architecture or a fully deployed lighting product, but a bounded software engineering evaluation of how architectural decisions shape the behaviour of an implemented perception-and-decision pipeline.

Within this scope, the study connects detector-level behaviour to control-relevant indicators such as missed occupied zones, delayed decisions, projected unnecessary lighting activation, processing-energy cost, and privacy-relevant data exposure. These indicators are not presented as evidence of realised building-scale operational impact. They show how architectural choices in the perception layer can propagate into the kinds of consequences that a deployed building-automation system would need to manage.

The practical significance of the work lies primarily in making these trade-offs explicit. By comparing implemented edge and cloud inference variants within the same zone-based control problem, the study makes trade-offs visible that may otherwise be hidden if detector accuracy, runtime performance, energy use, and privacy exposure are evaluated separately. The lighting model further allows projected application-level consequences to be compared against simpler baseline strategies, while keeping the actuation-level assumptions explicit.

The findings do not establish a universally superior deployment strategy. Instead, they clarify how different architectural choices expose different strengths and constraints in one application-specific setting. In that sense, the study contributes to research by providing a traceable evaluation workflow for perception-driven automation, and to practice by offering a structured basis for reasoning about inference placement under timing, energy, robustness, and privacy constraints.

The remaining chapters are organised as follows. Chapter 2 introduces the technical and application context, Chapter 3 positions the study relative to earlier literature, and Chapter 4 defines the conceptual framework used in the analysis. Chapter 5 explains the study design and evaluation procedure, Chapter 6 presents the results, Chapter 7 interprets those results in relation to the research questions, and Chapter 8 summarises the main conclusions and outlines future work.

2

Background

This chapter provides the technical and domain context needed to understand the study. It introduces the application setting, sensing concepts, deployment architectures, and system constraints used later in the study. Its purpose is explanatory, not comparative: the chapter defines the technical foundations of the studied system, while Chapter 3 reviews and compares prior research studies.

2.1 Building-automation and lighting context

Building automation systems coordinate building services such as heating, ventilation, air conditioning, and lighting through combinations of sensing, decision logic, and actuation. In public and commercial buildings, lighting is one of the controllable services through which operational efficiency measures can affect electricity use [2]. Lighting control can range from manual switching to automated strategies based on occupancy, schedules, daylight availability, or combinations of these inputs. In automation terms, lighting provides a bounded control problem where sensed information is translated into visible actions in the environment. The quality of the control signal affects when and where lighting is activated, which makes sensing accuracy, timing, and spatial resolution relevant to system behaviour.

2.2 Zone occupancy as a control signal

Occupancy can be represented at different levels. Some systems only need to know whether anyone is present. Others may need a count of how many people are present, for example in HVAC-related applications. Zone-based lighting control, however, requires spatial information about *where* people are in the monitored area, because the controller must determine which predefined part of the space should be lit.

Here, a *zone* is a predefined spatial region associated with one or more light fixtures or lighting states. The primary control signal is *zone occupancy*, that is, whether a given zone is occupied or unoccupied. In-zone counts may still be informative, but they are secondary to the occupancy decision itself.

This distinction matters because different errors have different control consequences.

A false positive can brighten an empty zone and waste energy. A false negative can leave an occupied zone too dark. Boundary cases are especially relevant, because small localisation errors near a zone border can cause the controller to alternate between adjacent zones. For that reason, the quality of the perception output is not judged only by frame-level detection correctness, but also by whether the resulting zone assignments are stable and usable as control decisions.

2.3 Conventional occupancy-sensing reference technologies

Conventional occupancy sensing in buildings can be implemented using several different sensing principles, including motion, distance, interruption, and radar-based detection. These technologies differ not only in response characteristics, but also in coverage, spatial selectivity, sensitivity to occupant movement, and suitability for zone-level control.

Academic literature is useful for classifying occupancy-sensing modalities and discussing evaluation protocols, but published timing information is often not directly comparable across sensor families. For example, Zhang et al. propose a protocol for evaluating occupancy-sensing technologies in building HVAC control, but their case study reports characteristics for specific implementations, not a standardised entry-to-detection latency across sensor types [11]. Table 2.1 summarises representative published timing fields and practical deployment limitations from concrete sensor examples. The table should be read as practical sensing context, not as a controlled benchmark comparison.

Table 2.1: Representative published timing fields and deployment considerations for conventional occupancy-sensing technologies.

Sensor type	**Published timing field**	**Practical sensing context**	**Source**
Through-beam photoelectric / light barrier	< 0.4 ms response time	Very fast interruption detection, but limited to a narrow beam between sender and receiver. More suitable for doorway or line-crossing detection than whole-room occupancy sensing.	[12]
Ultrasonic distance sensor	30 ms response time, 8 ms output time	Fast short-range ranging, but the cited device has a limited operating range and depends on target reflection, size, and alignment. Room-scale use would require additional system design.	[13]
60 GHz ceiling-mounted radar	150.1 ms frame repetition time	Supports ceiling-mounted room-scale sensing and can detect moving or stationary occupants. The reported value is a configured update period, and accuracy may vary across the field of view.	[14]
Passive infrared (PIR)	1 s response/-fade to switch on/off	Common in lighting control, but motion-based. Coverage can include blind spots, and idle occupants may stop triggering after the hold period expires.	[15]

Sensor type	Published timing field	Practical sensing context	Source
Microwave motion sensor	No direct trigger latency stated, minimum configurable hold time 5 s	Provides broad coverage and can detect through some non-metal materials, but the cited datasheet does not state direct trigger latency. The 5 s value is a configurable hold time, not a sensor reaction time.	[16]

The timing fields in Table 2.1 are heterogeneous: some describe sensor response time, some describe update periods, and some describe lighting-control behaviour or hold-time configuration. They are used only as contextual reference points for understanding practical sensing constraints, not as equivalent latency measurements.

2.4 Stereo-depth based spatial assignment

A 2D detector can indicate that a person appears somewhere in an image, but zone-based lighting control requires the detection to be related to physical space. Stereo-depth sensing supports this by adding distance information to the image observation, making it possible to estimate where a detected person is located relative to predefined control zones [17, 18]. Conceptually, the process consists of associating a person detection with depth information, projecting the resulting estimate into a scene coordinate representation, and testing that position against zone boundaries. This does not require full scene reconstruction, but it does require calibrated geometry and a stable mapping between camera observations and zone definitions.

The use of spatial information in smart lighting is established in prior work. Chun et al. describe real-time smart lighting control using human motion tracking from a depth camera [7], and Chun and Lee present smart lighting control as one application of human motion tracking [8]. Li et al. further show that stereo vision can be used in intelligent lighting systems [19]. These studies motivate the use of visual geometry for lighting-related control, but they do not focus on the deployment question addressed here: how the perception pipeline behaves when deployment and detector choices are varied within the same zone-occupancy task.

The system uses a Stereolabs ZED 2i stereo camera, which provides aligned visual and depth information suitable for mapping image-based person detections to spatial control zones [20]. In the implemented architecture, depth extraction and zone assignment remain edge-side for all evaluated deployment variants, while person-detection inference is the component varied between edge and cloud execution. Further implementation details of the spatial-assignment procedure are described in Chapter 5.

Depth support does not remove all uncertainty. Zone assignment can still be affected by calibration quality, depth noise, occlusion, and imperfect correspondence between a detected body position and the spatial point used for control. Depth accuracy

under indoor conditions has also been studied directly for the ZED 2i platform used here [18]. These uncertainty sources motivate the later evaluation of zone-occupancy accuracy, missed-zone behaviour, and robustness under more difficult visual conditions.

2.5 Deployment architectures for continuous perception

Deployment architecture describes where parts of a perception pipeline execute and how data and results move between system components. For continuous perception, this placement affects the compute resources available to the model, the communication path between sensing and inference, and the timing behaviour of the overall pipeline. Prior edge-computing work shows that object-detection performance, throughput, and resource demand are closely linked on constrained devices [4], while offloading work shows that moving computation between local and remote execution can shift energy and timing behaviour [9]. For camera-based systems, placement also affects what visual data must leave the sensing device, which connects deployment architecture to the privacy constraints discussed in Section 2.6.2 [5].

2.5.1 Edge inference

In an *edge* design, inference runs on the device attached to the camera or close to it. This avoids a remote inference request during normal operation and keeps processing near the sensing source. Edge execution can also reduce the need to transmit raw or derived visual data outside the local device. The main constraint is that model selection and runtime configuration must fit within the compute, memory, power, and thermal limits of the local hardware.

2.5.2 Cloud inference

In a *cloud* design, frames or derived representations are sent to a remote endpoint for inference. This can provide access to stronger compute resources and reduce the inference burden on the local device, but it introduces dependence on communication delay, request handling, network variation, and remote-service availability.

The cloud design evaluated here is a partial-offloading architecture. Person-detection inference is offloaded, while stereo-depth extraction and zone assignment remain on the edge device. This is partly a technical consequence of the locally attached ZED camera and its runtime coupling, and partly a way to keep the comparison focused on the placement of the detector inference stage.

2.5.3 Cloud execution environment

A cloud execution environment provides remote compute, storage, and networking services that can be used to offload parts of an application from an edge device. The

relevant background point is architectural: cloud inference depends on model execution, external infrastructure, and communication paths. Provider-specific service comparison is outside the study scope.

2.5.4 Model size and deployment feasibility

The deployment question is closely related to model configuration because continuous perception requires inference to be performed repeatedly over extended periods of time. Prior edge-device benchmarks of YOLOv3, YOLOv5, and YOLOX variants report detection quality, inference time, power consumption, and feasibility-oriented ratios such as FPS per watt on devices including the Jetson Nano [4]. Model choice is treated as part of deployment feasibility, not as a separate algorithmic concern.

Here, deployment feasibility refers to whether a model and placement configuration can support sustained operation within the available hardware, communication, timing, and energy boundaries. In the evaluation, this is operationalised through decision-path timing, runtime counters, measured power, and processing-energy estimates.

2.6 System constraints

A camera-based perception system in building automation is not judged only by whether it produces correct detections. It must also produce outputs in a way that is suitable for the surrounding control task. Two constraints are especially relevant here: *response quality*, which concerns the timeliness and stability of control-relevant outputs, and *privacy*, which concerns how person-related visual data is collected, transmitted, stored, and processed.

2.6.1 Response quality

For a zone-occupancy controller, response quality concerns how reliably and promptly occupancy changes are reflected in the control signal. A system may perform well on individual frames but still behave poorly as a controller if delay, stale results, or unstable zone assignments lead to late or repeated projected lighting transitions.

Here, response quality is introduced as a background concept and defined operationally in Chapter 5. The empirical latency measurements focus on the path from frame capture to occupancy decision, while actuator-side switching delay is represented as a constant in the control model. Robustness is evaluated through scenario conditions that stress the perception pipeline, such as obstruction and illumination variation.

2.6.2 Privacy constraints

Camera-based sensing introduces privacy constraints because captured images and related metadata may contain person-related information. Kansal et al. discuss how

privacy regulation affects video-surveillance-system design and influences architectural choices [5]. A central architectural distinction is whether visual data remains local or must be transmitted to an external service. Edge inference can keep more processing on the sensing device, whereas cloud inference may require frames or derived representations to leave the local environment.

Privacy-preserving tracking work also shows that sensing and representation choices can be shaped by privacy requirements. For example, Song et al. present a privacy-preserving human-tracking approach in ambient assisted living [21]. Although the sensing modality differs from the one used here, it illustrates the broader point that privacy constraints can affect which sensing pipelines and deployment architectures are considered acceptable.

3

Related Work

This chapter reviews and compares prior studies most directly related to the study. Whereas Chapter 2 introduced the technical setting and key concepts, the purpose here is to position the study relative to existing literature. The review is organised around literature streams that map directly to the research problem: AI in building automation, depth- and vision-based smart-lighting systems, efficient deployable AI, inference-placement trade-offs, and privacy-related architectural constraints.

3.1 AI for building automation and energy-efficient lighting

A first stream of related work concerns AI- and data-driven approaches to building automation and energy efficiency. Ortiz-Peña et al. show that operational improvements and improved control can be meaningful levers for reducing electricity use in public buildings [2]. This provides an energy-efficiency motivation for smarter control strategies, but it does not by itself indicate how a camera-based control loop should be designed.

Work closer to deployment is represented by Xie et al., who discuss how AI can be integrated into building automation systems at the field level [3]. Their contribution is relevant because it shifts attention from model quality alone to system feasibility, including limited field hardware, heterogeneous device ecosystems, and operational robustness. At the same time, their discussion is general with respect to sensing modality and control task. It does not evaluate a continuous camera-based zone-occupancy controller with explicit edge/cloud deployment trade-offs.

Together, these studies justify the application setting and its practical relevance, but they do not resolve the central comparative question studied here: how a spatially aware zone-occupancy controller should balance perception quality against latency, compute demand, and deployment constraints. This stream supports *why* lighting is an appropriate evaluation setting, but contributes less directly to the core questions about inference placement and control-loop consequences.

3.2 Depth- and vision-based smart-lighting systems

A second literature stream concerns lighting-control systems that use richer sensing than traditional binary occupancy detection. This stream is relevant because it shows that visual, depth-based, and geometry-aware representations can support more spatially specific lighting decisions. The studies summarised in Table 3.1 demonstrate different ways in which spatial sensing has been connected to lighting-related control or measurement.

Table 3.1: Depth- and vision-based lighting studies relevant to spatially aware control.

Study	**Sensing / representation**	**Lighting-related focus**	**Relevance to this study**
Chun and Lee [8]	Human motion tracking with visual sensing	Smart-lighting control as an application of motion tracking	Shows that lighting control can be driven by person-related spatial information, but does not treat inference placement as an architectural comparison variable.
Chun et al. [7]	Depth-camera-based human motion tracking	Real-time smart-lighting control using occupant position and activity information	Demonstrates depth-informed sensing for lighting automation, while using a largely fixed sensing-and-compute setup.
Słomiński and Sobaszek [22]	Dynamic identification and depth-related processing of moving objects	Intelligent lighting of moving objects with discomfort-glare limitation	Connects processing time and spatial sensing to visible lighting behaviour, but is not framed as an edge–cloud deployment study.
Ning et al. [1]	Video-based multi-occupant positioning	Fine-grained modelling and optimisation for multi-occupant smart lighting	Shows the value of richer spatial information for fine-grained control, but focuses on lighting optimisation once positions are available, not on deployment of the perception layer.
Li et al. [19]	Binocular stereo vision	Illuminance measurement for intelligent LED lighting	Supports the use of stereo-derived geometry in lighting applications, but does not study occupancy-driven zone control.
This study	Stereo-depth-assisted person detection and zone assignment	Zone-occupancy decisions for projected lighting-control behaviour	Evaluates deployment and detector choices within the same zone-based perception-and-decision pipeline.

Taken together, these studies support the use of spatial sensing in lighting-related systems, but they leave open the deployment question addressed in this study. Prior

work generally demonstrates that richer spatial information can support lighting control or lighting-related measurement. It does not sufficiently examine how the same zone-occupancy task behaves when inference placement, detector configuration, and resource constraints are treated as explicit comparison variables.

3.3 Efficient AI and embedded detection on constrained hardware

A third literature stream focuses on the resource demands of running AI continuously on constrained platforms. This stream is important because the system studied here is not a one-off inference task, but a long-running perception workload embedded in a control loop.

Miller et al. review lightweight AI methods such as pruning, quantisation, and knowledge distillation, showing how models can be adapted to tighter compute and memory budgets [23]. This literature is relevant because it provides a broader rationale for treating model configuration as a deployment variable, not as a purely algorithmic choice.

Dikshit et al. provide a recent example of knowledge distillation being used to reduce model size while retaining much of a larger model's capability [24]. Their work is in multilingual language modelling, not object detection, so it is not used here as domain evidence for smart lighting or YOLO deployment. Instead, it supports the more general rationale for including distilled model variants when the engineering problem involves balancing model capability against runtime demand.

Lema et al. are central here because they benchmark object-detection models on constrained edge platforms such as Jetson-class hardware [4]. Unlike the broader efficient-AI literature, their study is directly about deployable detection pipelines. It shows that performance depends not only on model family and size, but also on the hardware target and optimisation path. The missing piece, relative to this study, is that the evaluation stops at detector-level metrics such as throughput, power, and efficiency. It does not ask how those properties affect the behaviour of an occupant-facing lighting controller.

This difference affects how detector benchmarks can be used. A deployment that looks attractive in terms of frames per second (FPS) or device-level power draw may still be unattractive if it produces stale zone assignments or unstable projected lighting-control decisions. The evaluation uses this literature as a benchmark-oriented reference point, but extends it towards control-loop consequences. In this context, those consequences include delayed or unstable zone assignments, degraded projected lighting behaviour, and the possibility that a detector-level improvement may not translate into better application-level behaviour.

3.4 Inference placement: edge and cloud trade-offs

A fourth literature stream concerns where inference should run. For this study, this stream most directly informs the comparison of implemented edge and cloud deployment strategies. At the exact intersection of smart lighting, camera-based occupancy reasoning, and deployment-placement comparison, however, the literature is relatively sparse. For that reason, adjacent offloading work becomes relevant as transfer evidence, not as application-matched evidence.

Nguyen et al. compare on-device computation with remote execution in another AI setting and show that remote approaches can reduce client-side energy consumption while increasing end-to-end delay [9]. The paper is not a building-automation study, and it is used here cautiously for that reason. Its value lies in making a transferable systems point explicit: moving computation away from the client can shift resource demand while worsening timing. Because this evidence comes from an adjacent, not application-matched, domain, it should be interpreted as indicating plausible architectural trade-offs, not directly predicting smart-lighting outcomes.

What remains open in the study context is how those trade-offs appear in a control loop whose outputs are immediately visible to occupants. A cloud configuration that looks attractive in terms of local energy consumption may still be poor if communication delay or timing variation produces stale lighting decisions. That end-to-end comparison is not directly answered by the adjacent offloading literature, which is why it remains a central motivation for the present study.

The empirical comparison is between edge and cloud person-detection inference under a shared controller pipeline. Due to camera and SDK coupling, depth extraction and zone assignment remain edge-side in the implemented edge and cloud configurations.

3.5 Privacy and observability in deployable camera systems

Not all relevant literature concerns the detector or control policy directly. Some work instead defines broader system qualities that affect whether a camera-based controller is practical to deploy, including observability, privacy, and timing-related aspects of control quality.

Mohammad et al. review explainable AI strategies for Internet of Things (IoT) systems and argue that interpretability can support trust, transparency, accountability, and more understandable AI-driven decisions [25]. Here, that point is used in a limited engineering sense: a deployable controller should make its decision path traceable through intermediate states and runtime telemetry, not only report final detection outcomes.

Privacy-oriented work is also relevant because camera-based automation changes what information is collected, represented, and potentially transmitted. Kansal et al. discuss privacy-by-design in video pipelines, while Song et al. illustrate a sensing approach that reduces privacy exposure by emitting only 3D coordinates [5, 21]. These studies support the broader point that privacy requirements can affect which sensing representations and deployment architectures are acceptable. However, they do not evaluate the same RGB-plus-depth lighting-control pipeline across edge and cloud deployment variants.

Timing is another cross-cutting quality. Nielsen's response-time heuristics are not used here as lighting-specific thresholds, but as general HCI support for the claim that delay can affect perceived responsiveness in interactive or user-facing systems [26]. Lighting-specific work provides the closer application context: Chun et al. and Słomiński and Sobaszek show that timing and transition behaviour matter when sensing outputs are translated into visible lighting behaviour [7, 22]. Together, these sources support treating latency as part of control quality, while leaving the full edge–cloud, zone-occupancy evaluation problem to this study.

3.5.1 GDPR implications for testing and deployment

The GDPR is relevant because a camera-based zone-occupancy controller may process personal data whenever people are identifiable, directly or indirectly, in captured images or associated metadata [27, 28]. In this context, the system is not only a sensing-and-control pipeline but also a data-processing arrangement whose collection, transmission, storage, and use must be justified.

Legal and regulatory guidance does not answer the technical research questions, but it constrains which deployment options are realistic. GDPR-oriented principles such as data minimisation, purpose limitation, limited retention, and access control are especially relevant when raw or derived visual data is used in workplace-like environments [27, 28, 29, 30]. The study-specific handling of participation, recording practice, storage, and retention is addressed in Chapter 5, while this section uses the GDPR literature to position privacy as an architectural constraint.

Deployment placement is directly implicated by these constraints. A local edge design can, in principle, reduce the need to transmit raw imagery, whereas cloud designs may require frames or derived representations to leave the local device. When cloud infrastructure involves actors or transfer paths outside the EEA, the issue may also extend to third-country transfer assessment and supplementary safeguards [31, 32].

For this study, the literature supports a cautious conclusion: GDPR should be treated less as a final compliance claim and more as a practical design constraint that shapes which edge or cloud configurations remain proportionate and realistic in a European deployment context.

3.6 Synthesis and positioning

The reviewed literature supports three main observations. First, smart-lighting studies show that richer spatial sensing can improve control, but they usually assume a fixed perception stack without comparing alternative inference placements for the same task [8, 7, 1]. Second, efficient-AI and embedded-detection studies clarify throughput, power, and feasibility trade-offs on constrained hardware, but they usually stop at detector-level outcomes without tracing their consequences through an occupant-facing control loop [4, 23]. Third, privacy- and observability-oriented work shows that deployment architecture matters beyond raw accuracy, yet it often studies other sensing representations or other application goals [25, 5, 21].

The gap addressed by this study is narrower and more concrete than a general "AI for buildings" problem. The study evaluates one fixed, depth-informed, zone-based perception-and-decision pipeline while varying inference placement and detector configuration. Its contribution is not component-level novelty, but a comparative assessment of how those design choices affect zone-occupancy quality, control-loop behaviour, runtime burden, energy consumption, and privacy-relevant architectural exposure within one coherent application setting. In this scope, actuation-level outcomes are projected from models, not measured from a physically deployed lighting-actuation subsystem.

4

Theory

This chapter presents the conceptual framework used to analyse the study. It defines the main constructs through which the studied system is interpreted: zone occupancy as an application-level inference target, stereo-depth as a basis for spatial reasoning, control-loop quality as more than detector accuracy, inference placement as an architectural trade-off, and privacy exposure as a property of system design. Together, these concepts explain why the evaluation considers not only whether persons are detected, but also whether the resulting control behaviour is timely, stable, feasible, and privacy-aware.

The function of this chapter is conceptual, not descriptive. It uses the technical setting and related work introduced earlier to define the lenses through which the later methodology, results, and discussion are interpreted.

4.1 From person detection to zone occupancy

Zone occupancy connects detector output to control-level consequences. The relevant question is not only whether a person appears in an image, but whether the perception pipeline produces a state that can reasonably be used by a zone-based controller.

Zone occupancy is a derived state, not a direct detector output. Bounding boxes and confidence scores are evidence, but the control-relevant state also depends on how that evidence is interpreted spatially. This distinction is important because uncertainty can enter at more than one point: the detector may miss a person, and the spatial assignment may place a detected person in the wrong control relation.

The evaluation shifts from raw detection accuracy alone to the quality of the inferred occupancy state. Within this scope, an occupancy estimate is useful when it remains meaningful under conditions that challenge either detection or spatial assignment. This explains why the evaluation later combines occupancy accuracy, latency, stability, and robustness instead of treating detector performance as a separate result.

4.2 Stereo depth as a basis for spatial reasoning

Stereo depth provides a way to transform image-space evidence into a spatial hypothesis that can be tested against control zones. A detection is not treated only as an object in an image, but as a possible occupant location in the controlled environment.

Depth matters here because it enables *spatial gating*. Depth information can support rejection of detections that are inconsistent with the controlled area, reduce ambiguity near zone boundaries or openings, and make it possible to interpret a detection as a location estimate, not only an image-space event.

Depth is a mediating representation between perception and control. It does not by itself answer the research questions, but it makes zone occupancy meaningful as an evaluation target: the system can be evaluated on whether spatially interpreted detections lead to useful zone states.

At the same time, the benefits of depth should not be overstated. Stereo depth is an estimated quantity, so its role is enabling, not determinative. If the spatial hypothesis is uncertain, that uncertainty can propagate into zone assignment and then into downstream control behaviour. Depth is treated as a mechanism for spatial reasoning, not as a guarantee of correct localisation.

4.3 Control-loop quality in occupant-facing automation

A camera-based zone-occupancy controller can be understood as a perception–control loop: sensing produces evidence about occupancy, inference interprets that evidence, and control logic maps the inferred state to lighting-control actions. In the control model used here, those actions are represented as binary on/off activations, not as dimming levels. Fixture-level actuation is outside the empirical evaluation and is analysed only through projected activation intervals.

This distinction is particularly important in occupant-facing automation. A system may achieve reasonable average detection performance while still behaving poorly as a controller if small fluctuations around a zone boundary produce repeated projected on/off changes. In a practical deployment, the binary occupancy signal should normally be mediated by an actuator-side transition policy, for example gradual fades or dimming ramps, so that occasional perception errors do not become abrupt visible changes. Such smoothing is outside the empirical scope of this study because the project did not have access to control the physical fixtures and did not measure how much power the fixtures would consume at intermediate light-output levels. Prior smart-lighting work similarly frames lighting actuation as part of the overall control task [7]. Related dynamic lighting research likewise shows that sensing and inference delays matter because lighting responses must remain aligned with moving targets and comfort constraints [22].

For this reason, the study treats *latency*, *stability*, and *robustness* as central constructs. Latency concerns how long it takes for a perceptual event to influence the projected lighting state. Stability concerns whether the controller avoids unnecessary oscillation in response to noisy or ambiguous inputs. Robustness concerns whether the overall loop continues to operate meaningfully under changes in scene conditions, model choice, or deployment mode. These constructs are application-level, not model-level: they describe the quality of the resulting behaviour, not just the quality of the detector.

These constructs connect the conceptual framework to the later evaluation. RQ1 concerns zone-occupancy accuracy, robustness, and end-to-end latency. RQ3 concerns projected lighting behaviour and net-energy outcomes at the application level. The concepts introduced here define why the evaluation does not interpret a detector as successful solely because it performs well on isolated inference metrics.

4.4 Inference placement as an architectural trade-off

Inference placement is an architectural property that couples several quality attributes, not a location choice in isolation. Moving inference changes not only where computation happens, but also how timing, runtime feasibility, resilience, energy use, and data exposure relate to one another.

Local execution and remote execution represent different forms of constraint. Local execution reduces dependence on a remote communication path, but it also makes the controller more dependent on the compute, memory, power, and thermal limits of the local device. Remote execution can relax some local resource constraints, but it introduces dependence on transport delay, request handling, and remote-service availability. Neither placement can be evaluated only through detector accuracy.

For a zone-occupancy controller, placement can change the quality of the state delivered to the control logic. A model that is more capable per frame may still be less useful if its results arrive too late or become stale. Conversely, a smaller local model may remain attractive if it produces sufficiently reliable states with lower timing and communication burden. Inference placement is part of the control problem, not merely an implementation detail beneath it.

Architectural comparison must be application-level. Placement should be assessed through the consequences it creates for occupancy quality, timing, runtime burden, energy, and privacy-relevant exposure together. This is why the study evaluates edge and cloud configurations through system-level outcomes instead of treating placement as a secondary runtime setting.

4.5 Privacy-aware architectural reasoning

Camera-based automation is not judged solely by technical performance. The way information is represented, processed, and transmitted affects whether a system is realistic to deploy in shared indoor environments. Privacy is relevant here not only as an ethical afterthought, but as a design-level property that constrains acceptable architectures.

A useful conceptual distinction is the difference between *localisation* and *tracking*. Localisation estimates where a person is at a given time, for example in terms of coordinates or zone membership. Tracking, by contrast, maintains continuity of identity across time, often through trajectories, persistent identifiers, or appearance-based association. Some intelligent lighting systems use tracking to support richer behavioural reasoning across time or across multiple cameras [7]. However, not every control task requires that level of persistence. For zone-based lighting control, it can be sufficient to estimate the current in-zone occupancy state without maintaining long-lived identity information.

This distinction matters because data minimisation connects privacy requirements to design choices such as limiting detail, retention, access, and the degree of identification [30]. Ning et al. present a video-based positioning approach for multi-occupant smart lighting in which spatial occupant information is used as the basis for fine-grained lighting control [1]. Song et al. similarly present a privacy-preserving sensing approach in which the system relies on an infrared-sensor-based representation, not identity-rich camera imagery [21]. Kansal et al. make the broader point that privacy obligations in video systems apply across the full pipeline, including collection, storage, processing, and transmission [5]. Taken together, these works suggest that privacy exposure is shaped by architectural choices such as where inference runs, what intermediate representations are retained, and whether raw video must leave the edge device.

Here, the implication is modest but important: privacy-aware design is not separate from technical system design. It influences which deployment strategies are feasible, which data representations are proportionate, and how much of the perception pipeline can be justified in a practical building context. This does not, by itself, determine which architecture is best, but it clarifies why privacy exposure must be considered alongside latency, robustness, and energy consumption.

4.6 Implications of the theoretical framework

The framework has five direct implications for the rest of the study. First, success is defined at the level of *zone occupancy*, not raw person detection alone. Second, stereo depth is treated as the mechanism that makes spatial assignment possible, while still being recognised as an uncertain measurement, not a perfect ground truth. Third, the studied system is understood as a perception–control loop, which makes latency, stability, and robustness application-level outcomes instead of secondary

implementation details. Fourth, inference placement is treated as an architectural decision that shapes responsiveness, runtime feasibility, resilience, and data exposure, with empirical evaluation on edge and cloud. Fifth, privacy is treated as part of system design, not as an external afterthought.

These implications explain why the study compares deployment strategies using multiple criteria instead of a single detector benchmark. They also provide the conceptual bridge to Chapter 5, where the constructs introduced here are operationalised through explicit factors, compared conditions, and evaluation measures.

5

Methods

This chapter explains how the study operationalises the research questions. It describes the studied system, the compared conditions, how the evaluation is carried out, and which measures are used to analyse the proposed controller variants. The chapter makes explicit what is varied, what is measured, and what is kept constant when the same zone-based lighting-control task is evaluated under different deployment conditions.

The chapter combines methodological framing with technical specification because the contribution is not a stand-alone detector benchmark, but the design and evaluation of an end-to-end perception-and-decision pipeline situated in a lighting-control context. The method explains both how the system is built and how the comparison is performed, while keeping clear that physical light-fixture actuation is treated through separate calculations instead of as part of the implemented runtime.

5.1 Research approach and study design

The study is structured as an applied systems study in which alternative deployments of the same designed artefact are evaluated under controlled and field-like conditions. This framing is consistent with software engineering research strategies that study designed artefacts in relation to practical research problems and that require trade-offs between measurement control, realism, and generalisability [33]. The artefact under study is a camera-based, stereo-depth-assisted, zone-occupancy controller. The experiment compares alternative deployments of the same underlying control task in order to study trade-offs between perceptual quality, timing, runtime burden, energy consumption, and practical feasibility.

The principal independent variables are inference placement and model configuration. Cloud communication and request-handling behaviour are treated as part of cloud operation, while lighting-actuation baselines are modelled separately from the implemented runtime modes in the empirical evaluation. Across the full study, the main dependent variables are zone-occupancy accuracy, robustness, decision-path latency, runtime burden, operational energy consumption, projected lighting behaviour, and projected net-energy outcomes. To keep the comparison interpretable, the study holds the monitored scene, camera placement, zone definitions, down-

stream logic, and detector family constant where feasible.

The overall design is structured to answer the three research questions through complementary forms of evaluation. RQ1 is addressed through measurements of zone-occupancy accuracy, robustness, and decision-path latency. RQ2 is addressed through runtime and energy-related measurements, including communication overhead where applicable. RQ3 is addressed through model-based comparisons against simpler baseline automation strategies because physical lighting actuation is not implemented in the evaluated prototype. Controlled trials are used where repeatability is necessary, while the field-like normal-working-day trace and replay-based analysis are used where practical realism or cross-configuration comparability would otherwise be difficult to obtain in the same environment.

5.2 System under study

The studied system is a camera-based, zone-aware lighting pipeline that combines perception and spatial reasoning in a lighting-control context. Its purpose is to detect people in the monitored scene, estimate whether they occupy predefined spatial zones, and produce zone-occupancy states that can be used as input to a separate lighting-control calculation. The system is evaluated under edge and cloud inference-placement strategies while the overall control logic is kept as constant as possible. This design makes it easier to attribute observed differences primarily to placement and detector choice, not to changes in controller policy.

5.2.1 Hardware and sensing

The edge controller platform is the NVIDIA Jetson Nano Developer Kit. NVIDIA's user guide documents the relevant developer-kit revisions and integration details used in practice, including JetPack OS support, Gigabit Ethernet, and power-supply options for embedded deployment [34]. This platform is suitable for the study because it represents a constrained but realistic target for continuous edge-side perception.

The sensing platform is a calibrated stereo-depth camera, the Stereolabs ZED 2i, which provides RGB frames together with depth information aligned to the image through the ZED SDK [17, 20]. In the studied system, depth is used as a geometric cue for zone membership, not for full scene reconstruction. More specifically, detections are associated with spatial points and tested against predefined zone boundaries so that image-space detections can be translated into control-relevant occupancy states. The suitability of the ZED 2i for indoor spatial reasoning is further supported by prior analysis of its depth accuracy under indoor operating conditions [18].

Person detection is implemented using the YOLOv5 object-detection family, based on the Ultralytics reference implementation [35]. YOLOv5 was selected because it was compatible with the project hardware, TensorRT workflow, and available

trained model variants. Multiple YOLOv5 variants are considered in order to expose a controlled trade-off between model capacity, runtime demand, and deployment feasibility. The same detector family is retained across the comparison so that placement effects are not confounded with detector-family differences. Prior edge-device studies show that YOLO-family deployment on Jetson-class hardware involves trade-offs among throughput, power use, and feasibility, even though the exact model choice remains dependent on the hardware, software stack, and task [4, 36].

In edge configurations, person detection, depth extraction, zone assignment, and logging all execute on the local device. In cloud configurations, only person detection is offloaded to a remote endpoint. Depth extraction and zone assignment remain local because the implemented pipeline obtains depth through the ZED SDK and ZED Python API from the locally connected camera. A different stereo-camera architecture could in principle move more of the spatial-processing pipeline, but that redesign was outside the available time and study scope. The chosen method keeps the ZED-dependent spatial stage local and uses the cloud comparison to isolate the placement of person-detection inference.

5.2.2 Perception and control pipeline

At each control step, the system performs the following sequence:

1. the camera produces an RGB frame together with depth information.

2. the inference pipeline produces person detections, including bounding boxes and confidence scores.

3. detections are converted into spatial points using depth information and tested against zone definitions.

4. per-zone occupancy is computed, for example as an occupied/unoccupied state and, where relevant, an in-zone count.

5. the resulting occupancy state is logged and used as input to later lighting-control calculations.

This structure operationalises the conceptual distinction between person detection and zone occupancy introduced earlier in the study. The detector does not directly control the lighting. Instead, detections are translated into control-relevant spatial states. The evaluation reflects the actual task of the system, namely zone-based occupancy reasoning in a lighting-control setting, not generic frame-level detection alone.

The evaluated runtime does not apply a separate fixture-side smoothing layer. Instead, the implemented system relies on fixed mode settings such as shared thresholds and, in selected modes, motion-gated inference. These settings are held constant within comparable experiments so that observed differences can be attributed

to model and deployment effects, not to changing runtime parameters.

5.3 Compared conditions

The comparison is organised around a small set of explicitly defined factors so that the resulting analysis remains interpretable.

5.3.1 Inference placement

Inference placement is the primary architectural factor in the study. Two configurations are considered in the empirical comparison:

- **Edge:** person detection, depth extraction, zone assignment, and logging are executed locally on the edge device.
- **Cloud:** person detection is executed remotely, while depth extraction, zone assignment, and logging remain local and operate on the returned detections.

This factor is central because it changes the trade-off between local resource use, communication dependence, timing behaviour, and privacy exposure.

5.3.2 Cloud inference endpoint

The cloud configuration uses a GPU-backed HTTP inference service, not a provider-specific managed service. The edge device sends selected RGB frames as JPEG payloads to a configured `/infer` endpoint using the `DATX05_CLOUD_INFER_URL` setting. Before transmission, frames are downscaled so that the longest side is at most 640 pixels, and JPEG quality is set to 60. The endpoint is served by a FastAPI application running under Uvicorn in a Docker container with a CUDA-enabled PyTorch runtime.

On the remote endpoint, each request is decoded, passed to a YOLOv5 v7 inference runtime on `cuda:0`, filtered to the person class, and returned as JSON containing the frame identifier, bounding boxes, confidence values, payload size, and timing fields. The cloud model uses an image size of 640, confidence threshold 0.40, IoU threshold 0.45, and a maximum of 300 detections. The endpoint serialises model execution with a single inference lock, while the edge-side cloud mode controls how many requests may be in flight. The measured cloud behaviour therefore includes not only remote model inference, but also the configured request path, frame encoding, decoding, queueing, stale-result handling, and transport telemetry.

5.3.3 Model configuration

Model configuration is varied within the YOLOv5 family so that detector complexity can be studied without also changing detector family. Holding the family constant reduces confounding and supports cleaner interpretation of deployment effects. When placement is compared directly, the detector family is held constant

across configurations. When model scaling is analysed, the comparison is reported as a within-family comparison, not as a detector-family comparison.

Table 5.1: Model and deployment configurations used in the empirical comparison.

Configuration	Model variants	Execution meaning	Reported use
Edge	V5x, V5l, V5l-kd, V5m, V5s, V5s-kd, V5n	Person detection runs locally on the Jetson Nano using TensorRT engines. Depth extraction, zone assignment, and logging are also local.	Occupancy, timing, working-day trace, and processing energy
Cloud	Same seven retained YOLOv5 variants	Person detection runs on the remote HTTP endpoint. The edge device sends JPEG frames and receives detections, while depth extraction, zone assignment, and logging remain local.	Occupancy, timing, working-day trace, and processing energy
Cloud in-flight tuning	Same cloud endpoint, grouped by maximum in-flight request setting	Request concurrency is varied using maximum in-flight settings of 1, 2, 3, 4, and 8.	Queue-drop, stale-result, and timing telemetry
Excluded checkpoints	`best_kd_m`, `best_kd_n`	Generated outputs exist, but they are excluded from the reported analysis to keep the retained comparison consistent.	Not reported in result tables or figures

5.3.4 Conditional deployment factors

Some experimental factors apply only to specific deployment modes. For cloud logging modes, the maximum number of in-flight requests is treated as an operational parameter because it affects queue drops, stale results, and timing variation. Communication delay and remote-service availability are interpreted through the available timing and queue telemetry, not through a separate network-impairment experiment. Edge inference serves as the local reference condition because it is not subject to the same communication path.

Lighting-control granularity is treated as an application-level factor, not a perception factor. Where granularity itself is not the object of comparison, it is held constant so that observed differences can more clearly be attributed to deployment and model effects.

5.4 Projected lighting-actuation model

This section defines how logged zone-occupancy states are translated into projected lighting behaviour and theoretical lighting-energy use. The runtime prototype produces zone-occupancy decisions and technical logs. It does not directly switch the physical fixtures during the evaluated runs. For that reason, lighting behaviour is reconstructed after the fact from the logged red, yellow, and blue zone states. The model applies deterministic activation rules to these logged states so that the same occupancy trace can be compared under a zone-aware controller, a room-level motion-detector baseline, and a physical-switch baseline. This separates the measured perception output from the later lighting-energy calculation and makes clear that the reported lighting energy is projected, not measured at the fixture.

5.4.1 Lighting-actuation mode definitions

The comparison uses three lighting-actuation modes. In the zone-aware mode, the red, yellow, and blue zones are controlled independently. A detection in one of these zones activates only the fixtures mapped to that zone, and that zone remains active until 15 seconds after the zone is last observed as occupied. The no-zone state is not treated as a lighting zone and does not activate fixtures in this model. In the motion-detector baseline, occupancy detected anywhere in the lamp-bearing part of the room activates all lamp-bearing zones together. The lights then remain active for eight minutes after the last detected activity, because this is the delay used by the lighting-control behaviour in the tested rooms. In the physical-switch baseline, all fixtures in the evaluated room model are treated as on for the full analysed duration. All three modes are modelled as binary on/off lighting states. Dimming is not included in the calculation.

5.4.2 Physical lighting setup: zones, fixtures, wattage assumptions

The lighting calculation uses the physical room zones selected for the energy comparison. The red and yellow zones represent work areas, while the blue zone represents the corridor area. The model uses the fixture counts and nominal lamp wattages from the tested part of the room: the red work area is represented by two 36 W commercial LED panel fixtures, the yellow work area by two 36 W commercial LED panel fixtures, and the blue corridor by four 36 W commercial LED panel fixtures. These assumptions are encoded in the analysis pipeline and are used to compute projected lighting energy from the reconstructed lighting states. The experiment does not directly measure the lamps during the test.

5.4.3 False-positive treatment

False-positive detections matter because they can activate lighting even when no lamp-bearing zone is occupied. The workflow keeps false-positive lighting time separate from ordinary occupied-zone lighting time. A false-positive interval is counted

when the logged state indicates activity outside the lamp-bearing red, yellow, and blue zones while those lamp-bearing zones are empty. This value is used to identify erroneous lighting time and, where applicable, the projected lighting energy associated with that error. The purpose is not only to count detector mistakes, but to connect those mistakes to the response-quality consequences they would have in a lighting-control setting.

5.4.4 Full-day trace and projected lighting analysis

The full-day trace refers to the field-like working-day recording used for the extended lighting and energy analysis. It is called full-day because the test was intended to represent a normal day of work in the monitored environment, not because it represents a 24-hour measurement. The trace is still evaluated as one bounded recording rather than as long-term operational evidence. Each logged zone-state change is treated as valid until the next logged change or until the end of the analysed window. The zone-aware mode is reconstructed by applying the per-zone 15-second hold rule independently to the red, yellow, and blue zones. The motion-detector baseline is reconstructed by treating activity in any lamp-bearing zone as room-level activity and then applying the eight-minute hold rule. The physical-switch baseline is reconstructed as a single interval in which all lamps are active for the entire analysed duration. For each interval, the active fixtures are translated into watts using the configured zone-to-fixture mapping, and accumulated energy is calculated over time. The same reconstructed intervals also support timeline overlays that show when each modelled lighting mode would be active.

5.5 Data collection and evaluation procedure

The study combines controlled trials, one field-like normal-working-day trace, and trace-based analysis. Controlled trials provide repeatable scenarios, while the field-like trace preserves a more natural temporal sequence from the monitored environment. Trace replay and log-based reconstruction then allow the same underlying sequence to be compared across models and between edge and cloud execution.

5.5.1 Controlled trials and field-like trace

In the controlled trials, recorded occupants follow scripted movement patterns through the predefined zones so that different model variants and deployment strategies can be compared on similar inputs. The scenarios include events that are particularly relevant to the research questions, such as entering and leaving zones, hovering near boundaries, partial occlusion, and the presence of multiple occupants. By constraining movement patterns and scene conditions, the study reduces environmental variation and improves the comparability of the resulting measurements. The evaluation treats the recorded people as part of the test scene, not as a sampled participant population, and no participant-level statistical claims are made.

The field-like recording complements the scripted trials with a normal-working-day

sequence from the monitored environment. Exact side-by-side repetition across configurations would not be possible during live ordinary use, so the recording is captured as an SVO trace and replayed to the controller under different deployment settings. This strengthens internal comparability while preserving a closer connection to realistic operating conditions than purely synthetic tests would provide.

5.5.2 Implemented empirical evaluation procedure

The empirical results reported in Chapter 6 are based on repeatable batch logging runs in edge and cloud modes, together with a separate cloud in-flight tuning analysis. For the replay-based occupancy experiments, each selected YOLOv5 variant is executed on the same prerecorded SVO sequences so that the visual input remains constant across repeated runs and across compute targets. Edge runs are repeated 20 times per model and cloud runs 10 times per model. The checkpoints `best_kd_m` and `best_kd_n` are excluded by the analysis pipeline so that the reported tables and figures remain restricted to the seven model variants retained consistently across the generated edge and cloud outputs.

The occupancy analysis is based on controller frame-change logs, not sparse summary counts alone. For each run, the logged state changes are parsed and converted into a per-frame reconstruction of observed zone occupancy. This makes it possible to evaluate the controller at the level of the time-varying control signal, not only at the level of isolated detection events.

To keep the comparison consistent across models, each scenario is analysed within a fixed manually defined frame window that is shared by all model variants in that scenario. Start and end frames are not adjusted per model. Instead, the same scenario window is used for all runs, and the reconstructed observed occupancy is compared frame by frame against a manually defined expected occupancy trace for that scenario. These expected traces were defined by the authors from the recorded scenarios and checked against the scenario descriptions before analysis. They should not be interpreted as independently validated ground truth. This introduces annotation uncertainty, especially near transition frames and zone boundaries, where the exact moment of occupancy change may be ambiguous. The reported occupancy metrics should be read as performance relative to the defined annotations, not as absolute measurement against an independent ground-truth system.

The reported occupancy scenarios are defined as follows:

- **Day, no obstruction:** frames 44–430, with expected occupancy `no_zone`=1.
- **Night, no obstruction:** frames 34–420, with expected occupancy `no_zone`=1.
- **Day, object obstruction:** frames 45–485, with manually defined intervals for the *behind obstruction* and *outside obstruction* phases.
- **Day, person obstruction:** frames 1–457, with phase boundaries at frames 35, 146, and 393.

In addition to these occupancy scenarios, the cloud evaluation includes an in-flight tuning study in which cloud logging telemetry is aggregated under several maximum in-flight request settings. This part of the evaluation is used to study queue drops, stale results, and timing behaviour under different cloud request concurrency configurations. Longer edge and cloud working-day trace runs support the projected lighting and power-related comparisons over an extended sequence intended to reflect a normal day of work in the monitored environment.

This procedure is intentionally conservative. Because the same manual frame windows and expected occupancy definitions are reused across all models within a scenario, differences in reported performance are intended to reflect model and deployment behaviour, not scenario-specific alignment choices. The trade-off is that fixed windows may penalise models differently near scenario boundaries if one model changes state slightly earlier or later than another. This is accepted in order to avoid model-specific scoring adjustments.

5.6 Evaluation methodology

This section explains how the empirical evaluation is carried out in practice and why it is structured in this way.

5.6.1 Evaluation workflow

For the replay-based occupancy experiments, the prototype is executed in repeatable batch modes that replay the same SVO recording across multiple models and multiple runs. Each run produces log output that contains occupancy state changes for the relevant zones and, in cloud modes, additional timing and queue telemetry. Custom analysis scripts then reconstruct a dense per-frame occupancy trace from these sparse change logs and compare that trace against a manually defined expected trace within a fixed analysis window.

Scenario-specific difficulty variants are handled through fixed partitions within each recording. In the object-obstruction scenario, separate intervals are scored for behind-obstruction and outside-obstruction behaviour. In the person-obstruction scenario, the analysis uses fixed boundary frames to separate phases such as no-zone before, the obstructed two-identification period, and no-zone after. These partitions make it possible to evaluate not only overall scenario performance, but also which part of the scenario causes degradation.

For the cloud in-flight analysis, the methodology is different because the purpose is not spatial occupancy scoring but cloud-runtime characterisation. Here the logs are grouped by `cfg_max_inflight`, and repeated runs are summarised using queue-drop counts, stale-result counts, and timing fields such as total cloud time, request time, transport time, and inference time. For the working-day lighting analysis, sparse zone-state logs are reconstructed into continuous intervals and exported as CSV tables. The main table reports projected lighting energy for each model under

the zone-aware controller and the two baselines. Separate sidecar tables report reconstructed lighting intervals, runtime timing summaries, power summaries, and net-energy balances.

Table 5.2: Empirical material reported in Chapter 6.

Scenario	Compute target(s)	Repetitions	Main reported output
Day no obstruction	Edge, cloud	20 / 10 per model	Zone-occupancy quality
Night no obstruction	Edge, cloud	20 / 10 per model	Zone-occupancy quality
Object obstruction	Edge, cloud	20 / 10 per model	Variant-based occupancy quality
Person obstruction	Edge, cloud	20 / 10 per model	Multi-identification occupancy quality
Cloud in-flight tuning	Cloud	15 per configuration	Queue and timing telemetry
Working-day trace	Cloud, edge	1 per model and target	Projected lighting behaviour and energy

5.6.2 Rationale for the evaluation design

The evaluation is structured in this way for several methodological reasons:

- replaying the same SVO sequence across models and compute targets ensures that the compared configurations see the same visual input,
- reconstructing per-frame zone occupancy from controller logs evaluates the actual control-relevant signal, not only raw detector outputs,
- fixed analysis windows and fixed expected traces avoid model-specific tuning of the scoring procedure,
- repeated runs per model make it possible to summarise variability instead of relying on single-run behaviour, and
- separating cloud in-flight tuning from occupancy scoring isolates queueing and transport effects that would otherwise be difficult to interpret inside aggregate accuracy numbers alone.

An additional advantage is that the methodology reduces reliance on long-term storage of raw video for scoring, since the reported metrics can be derived from logged controller outputs and predefined scenario definitions. This supports both repeatability and data minimisation.

5.7 Evaluation measures

The evaluation measures are selected to match the control task and the three research questions. The study does not rely only on detector-level benchmarks, but evaluates the system at the level of zone occupancy, controller behaviour, runtime burden, and energy-related consequences.

5.7.1 Zone-occupancy measures

The primary performance measures are derived from frame-level comparison between reconstructed observed occupancy and expected occupancy. *Correct %* is the proportion of analysed frames in which the observed occupancy exactly matches the expected occupancy. *Missed %* is the complementary proportion of analysed frames that do not match the expected occupancy.

In addition to frame-level correctness, the analysis reports scenario-specific error terms. In count-based analyses, *drops* count missing individuals as $\max(\text{expected} - \text{observed}, 0)$. In binary presence analyses, drops instead describe interruptions in an expected occupied state, such as transitions from detected to not detected inside an expected-presence interval. *False positives* count excess individuals as $\max(\text{observed} - \text{expected}, 0)$ where count-based precision is meaningful. *Precision* and *recall* are then computed from aggregated true-positive, false-positive, and false-negative counts over the analysed frames where those terms are supported by the scenario definition.

These measures are used because the relevant output of the system is not generic image detection, but a control-relevant estimate of whether specific zones are occupied and, where relevant, how many occupants are present in them. In the simpler no-obstruction and object-obstruction analyses, precision is less informative than missed frames, drops, and recall because the task is dominated by expected-presence reconstruction. In the person-obstruction analysis, precision and recall are both retained because multi-person detection behaviour makes both under-counting and over-counting relevant.

5.7.2 Robustness and control-loop measures

Robustness is assessed by examining how performance changes under more difficult but relevant conditions, including illumination change, partial occlusion, boundary cases, and more crowded scenes. For cloud-related evaluation, robustness also includes communication-sensitive behaviour such as queue drops and stale results. This makes it possible to study whether a deployment alternative remains useful only under favourable conditions or also under more challenging ones.

Latency is treated as decision-path latency, not fixture-switching latency. The reported timing analysis uses the timing fields that are explicitly logged by the implemented runtime. Edge working-day logs provide local `request_ms` and `infer_ms` values, while cloud logs provide fields such as `cloud_total_ms`, `request_ms`, `transport_ms`,

`decode_ms`, and queue-wait timing, together with queue-drop and stale-result telemetry. These fields characterise the observed decision path, but they do not claim to isolate every possible capture and postprocessing contribution. Physical actuator delay is not part of this measurement because fixture actuation is not implemented in the evaluated runtime.

Stability is assessed indirectly in the reported empirical evaluation, primarily through occupancy misses, drops, stale-result behaviour, and the consistency of the reconstructed control-relevant signal. The prototype was not connected to any physical light fixtures during evaluation, and does not report any separate stability aggregate. The study can identify indicators that may affect stability, such as missed states or stale cloud responses, but it cannot directly measure visible fixture oscillation, dimming smoothness, occupant-perceived stability, or actuator-side behaviour. A dedicated stability evaluation would require fixture-level actuation or a comparable test setup where projected state changes can be observed as physical lighting responses.

5.7.3 Runtime burden and energy

Runtime burden is characterised through the telemetry that is available from the implemented system: decision-path timing, inference timing where separately logged, queue drops and stale results where cloud communication is involved, and throughput or frame-processing rate where it can be derived from the logs. These measures describe whether a deployment configuration can produce useful occupancy decisions continuously, without promising direct CPU utilisation, GPU utilisation, or memory-footprint measurements. Edge working-day runs do not use the cloud request queue, so cloud-specific sent, dropped, stale-result, and failure counters are not meaningful for that mode.

Energy-related measures connect the processing cost of the sensing and inference pipeline to the projected value of lighting control. Processing energy is calculated from available power telemetry, while lighting energy is calculated from reconstructed lighting intervals. To keep these terms explicit, the study also reports a net-energy perspective:

$$E_{\text{net}} = E_{\text{lighting saved}} - E_{\text{processing}}$$

This formulation makes explicit that improved lighting control should be interpreted together with the energy use of the sensing and inference pipeline itself. In the reported calculations, $E_{\text{processing}}$ is the measured edge total energy for edge runs and the combined measured edge-side plus cloud-GPU energy for cloud runs. The lighting-saved term is a modelled projection, not a directly measured fixture-level saving.

5.8 Energy consumption

This section documents the power-logging implementation and measurement boundaries behind the processing-energy terms introduced above. Runtime energy is treated separately from projected lighting energy so that the sensing and inference pipeline is not hidden inside the lighting-control comparison. Where direct power samples are available, average power is converted to energy over the measured runtime.

5.8.1 Power-logging instrumentation details

This subsection documents the instrumentation used for power-related measurement and logging. On the edge device, Jetson power telemetry is sampled with `tegrastats` at a 1000 ms interval. The analysis uses total device input power as the processing-energy basis. GPU power is retained only as diagnostic information and is not added again.

For cloud inference, direct whole-system power measurement on the remote infrastructure is not available. The analysis combines logged cloud-GPU power with edge-side total input power and converts average power over the analysed duration to processing energy. Network routers, full cloud-host power, CPU, memory, storage, and datacentre overhead are excluded because they cannot be measured reliably within the study setup and are outside the study's measurement boundary.

5.8.2 Energy-assumption boundaries

These boundaries mean that the cloud results should be read as partial processing-energy estimates, not as complete cloud-infrastructure energy. Cloud provider billing is not analysed as a direct financial charge. The calculation also keeps measured processing energy separate from projected lighting values and from explicitly excluded components.

5.9 Baseline automation strategies

To address the application-level value of the proposed controller, the system is compared with simpler baseline strategies that represent different levels of lighting-control granularity. These baselines act as modelled comparators for interpreting the practical value of depth-aware zone reasoning.

The baseline strategies are:

- **Physical-switch baseline:** all mapped fixtures in the tested room model are treated as active for the full analysed duration.
- **Motion-detector baseline:** activity anywhere in the lamp-bearing part of the room activates all mapped fixtures, and the active state remains for eight

minutes after the most recent detected activity.

- **Zone-aware controller:** each lamp-bearing zone is controlled independently from its reconstructed occupancy state, so that the red and yellow work areas and the blue corridor can be lit separately.

These baselines represent different levels of lighting-control granularity. The physical-switch baseline captures the energy consequence of keeping the tested fixture group active throughout the evaluated period. The motion-detector baseline captures the behaviour of room-level occupancy-triggered lighting with the same eight-minute delay used in the tested rooms. Comparing these baselines with the zone-aware controller helps determine whether the added complexity of depth-assisted zone reasoning produces visible benefits at the application level, especially in relation to projected lighting behaviour and projected energy outcomes. These baselines are treated as modelled comparators, not as separately measured fixture-control experiments.

5.10 Ethical and privacy considerations

Because the studied system uses person-related visual data, ethical handling is treated as part of the method, not only as a later discussion point. In addition to GDPR-related requirements, the study is guided by research-integrity principles of reliability, honesty, respect, and accountability [37, 27].

5.10.1 Consent and recording practice

Participation in recordings is voluntary and based on informed consent, with particular care taken because the recordings take place in a workplace-like setting [29]. Before recording, participants receive written information about the purpose of the study, what data is collected, how the material will be used, who may access it, and how long it will be retained. Participants are informed that they may withdraw, and identifiable material associated with a withdrawal is removed from analysis where feasible.

Recordings are limited to defined test areas and planned recording periods. Visible notice is used when the camera is active, and material that unintentionally captures non-participants is excluded from analysis and deleted as soon as practicable. The collected material is used for study evaluation only, primarily through controller logs, derived occupancy states, and aggregated results instead of long-term retention of raw imagery.

5.10.2 Data minimisation, storage, and retention

The study is designed so that raw imagery is not required for the primary evaluation. By default, the retained material is limited to what is needed for the reported metrics, such as derived controller outputs, technical logs, and aggregated analysis

results. This follows the data-minimisation logic of limiting detail, retention, access, and identifiability where possible [30]. The system does not perform face recognition, identity inference, or long-term person tracking. Occupancy is treated as an identity-free control signal.

Stored data is handled according to the principle of least privilege. Access is restricted to the authors and, where necessary, supervisors or examiners. Where storage or transfer is required, the study uses controlled project storage and encrypted channels where feasible. Raw frames collected for annotation are kept only as long as needed for labelling and verification and are then deleted. Derived logs and aggregated results are retained only for the duration required to complete the project and examination, after which they are removed or archived in anonymised form where appropriate.

5.10.3 Privacy-aware deployment reasoning

Because inference placement is one of the main variables in the study, the evaluation explicitly accounts for how privacy exposure differs across configurations. Edge inference is preferred where feasible because it avoids transmitting raw camera data beyond the local device. When cloud inference is evaluated, the study treats data transfer and persistence as design constraints: the intended role of the cloud endpoint is transient inference, not long-term storage, and any broader deployment would need to consider infrastructure location, transfer paths, and supplementary safeguards where applicable [31, 32].

These choices operationalise the study-wide claim that privacy is a design constraint. They also ensure that the comparison between edge and cloud deployment is not interpreted only as a performance comparison, but also as a comparison of architectural exposure.

5.10.4 Research integrity and reporting practice

The European Code of Conduct for Research Integrity emphasises reliability, honesty, respect, and accountability as core principles for responsible research practice [37]. In this context, these principles are operationalised through documented experiment scripts, transparent reporting of methodological constraints, proportionate handling of participant-related data, and explicit responsibility for storage, deletion, and analysis. This framing helps keep the ethics section concrete and makes clear that responsible handling of participants, data, and reporting quality is part of the methodological design of the study.

5.11 Threats to validity

This section identifies the main validity threats associated with the study and how they are addressed.

5.11.1 Internal validity

Internal validity concerns whether observed differences can reasonably be attributed to the studied factors, not to uncontrolled confounds. Here, threats include calibration errors, annotation bias, variation between experimental runs, and unintended differences in configuration between deployment variants. These threats are mitigated through the ZED 2i's calibrated sensing pipeline, predefined zone definitions for each scenario, repeatable trial design, consistent logging, and repeated measurements across comparable scenarios.

5.11.2 External validity

External validity concerns how far the findings may transfer beyond the studied setting. The results may depend on the layout of the room, the camera placement, the lighting conditions, the distance to occupants, and the specific characteristics of the tested hardware and network environment. For that reason, the study interprets its conclusions within the scope of the evaluated environment and reports the scenario characteristics needed to understand those limits.

The field-like material should be interpreted as illustrative of one monitored environment, not as representative of all workplaces, building types, or geographical regions. Generalisability is also affected by the data used to train the detector variants. Because the study uses available trained models and open-source training material instead of site-specific long-term data collection, performance may differ in environments with other clothing, lighting, camera angles, furniture layouts, or occupancy patterns. A deployed system could potentially improve local robustness by collecting and retraining on data from its own environment, but this would introduce additional privacy, consent, retention, and GDPR-related constraints. For that reason, the study uses limited controlled and field-like recordings for evaluation instead of continuous self-training on workplace data.

5.11.3 Construct validity

Construct validity concerns whether the chosen measures capture the qualities the study claims to evaluate. The metrics are chosen to match the purpose of the controller: zone-occupancy accuracy, robustness, decision-path latency, stability, runtime burden, and energy-related impact. A remaining limitation is that cloud-related energy outcomes do not represent complete provider-side infrastructure energy. The results are interpreted within the stated measurement boundary, not as complete datacentre-energy measurements.

The expected traces were manually defined by the authors from the recorded scenarios and then checked against the scenario descriptions before analysis. They should not be interpreted as independently validated ground truth. This creates a risk of annotation bias, especially near transition frames and zone boundaries, where the exact moment of occupancy change may be ambiguous.

5.11.4 Reliability

Reliability concerns whether the study could be repeated with the same procedure and produce comparable observations. To support reliability, experiments are scripted where possible, the evaluation pipeline is logged, and the main analysis steps are documented. This does not eliminate all uncertainty, especially in field-like settings, but it improves the traceability and repeatability of the study.

6

Results

This chapter presents the empirical and analytical results used to answer the three research questions. The first part reports zone-occupancy accuracy and robustness under the controlled replay scenarios. The second part reports decision-path latency and cloud runtime-burden telemetry. The third part reports full-day projected lighting behaviour, processing energy, and net-energy outcomes relative to the baseline automation strategies.

The results should be read with the measurement boundaries defined in Chapter 5. The controlled occupancy results are based on repeatable replay runs. The full-day lighting results are analytical projections from logged zone states rather than measurements from physical fixture actuation. Edge full-day runs do not use the cloud request queue, so cloud-specific sent, dropped, failure, and stale-result counters are only meaningful for the cloud-mode full-day run. Cloud energy includes logged cloud GPU power and measured edge-side power, while full cloud-host power, CPU, memory, storage, network equipment, and datacentre overhead are outside the measurement boundary.

6.1 Zone-occupancy accuracy and robustness

This section addresses the accuracy and robustness part of RQ1. The main reported metric is recall, because missed occupied states are especially important for a zone-occupancy controller: if an occupied zone is not reconstructed as occupied, the projected lighting-control model may fail to activate the relevant lights. Precision is also reported where multi-person behaviour makes over-counting relevant.

6.1.1 No-obstruction scenarios

The no-obstruction scenarios provide the least occluded comparison setting and therefore serve as a reference point for how the evaluated models behave under simpler daytime and night-time conditions. Table 6.1 reports the mean recall across models together with the best- and worst-performing model for each deployment target.

Across both no-obstruction conditions, cloud inference achieved higher mean recall

Table 6.1: No-obstruction results. Mean recall is averaged across all evaluated models for the stated condition, with sample standard deviation reported in percentage points.

Scenario	Target	Mean recall ± SD	Best	Worst
Day no obstruction	Edge	91.50% ± 8.87	V5x (97.6%)	V5s (74.3%)
Day no obstruction	Cloud	98.54% ± 0.79	V5l / V5x (99.5%)	V5s-kd (97.5%)
Night no obstruction	Edge	56.39% ± 9.61	V5s-kd (69.1%)	V5s (39.5%)
Night no obstruction	Cloud	92.69% ± 3.44	V5s (96.9%)	V5l-kd (88.5%)

than edge inference. The difference was modest in the daytime setting, where the strongest edge models were close to the cloud models, but it became large at night. The night-time result functions as an illumination-robustness result even though no object physically obstructed the person: reduced illumination was associated with substantial degradation in the edge deployment.

6.1.2 Object-obstruction scenario

The object-obstruction scenario isolates performance under partial visual obstruction and is therefore reported separately for the behind-obstruction and outside-obstruction phases. Table 6.2 summarises the mean recall for each variant and the best-performing model in edge and cloud execution.

Table 6.2: Object-obstruction results by scored variant.

Variant	Edge mean	Cloud mean	Best edge	Best cloud
Behind obstruction (zone1 > 0)	40.94% ± 8.77	81.39% ± 0.56	V5x (52.4%)	V5s (82.1%)
Outside obstruction (no_zone > 0)	84.93% ± 7.75	99.50% ± 0.67	V5l-kd (94.4%)	Several models (100.0%)

In both scored variants, cloud inference produced higher mean recall than edge inference. The largest absolute gap appeared in the behind-obstruction phase, where the edge results degraded more strongly. This phase is relevant to lighting control because real rooms contain furniture, people, and other visual barriers that can partially hide occupants. A controller that performs well only when the person is fully visible may still produce missed or delayed lighting decisions in ordinary use.

6.1.3 Person-obstruction scenario

The person-obstruction scenario introduces multi-person detection behaviour and is therefore reported both for the main zone-focused variants and for the no-zone-focused variants. Tables 6.3 and 6.4 summarise the corresponding recall results.

Table 6.3: Person-obstruction results for the main zone-focused variants.

Variant	Edge mean	Cloud mean	Best edge	Best cloud
Combined total (zone1+no-zone)	87.03% ± 5.60	92.79% ± 4.30	V5n (95.2%)	V5m (96.6%)
Zone1 total	84.51% ± 6.67	91.26% ± 5.41	V5n (94.1%)	V5m (96.1%)
Zone1 person_obstructed	78.39% ± 8.98	91.19% ± 7.71	V5n (91.5%)	V5m (98.1%)
No-zone total (before+after)	97.09% ± 2.48	98.86% ± 0.35	V5n (99.7%)	V5n (99.4%)

Table 6.4: Person-obstruction results for the no-zone focused variants.

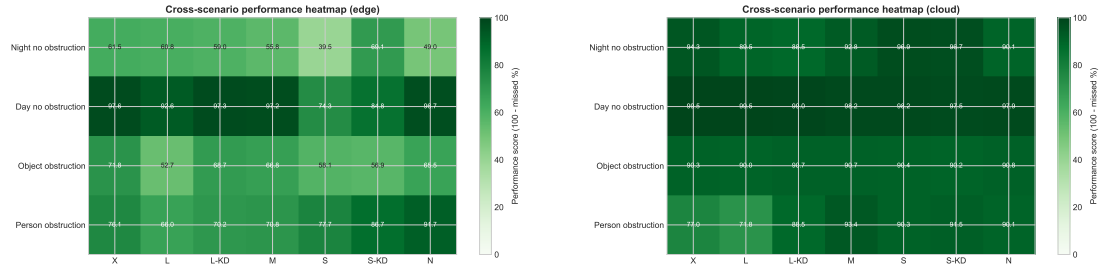
Variant	Edge mean	Cloud mean	Best edge	Best cloud
No-zone before	97.87% ± 1.70	98.47% ± 0.59	V5s (99.8%)	V5l / V5n (99.0%)
No-zone after	95.76% ± 3.96	99.46% ± 0.72	V5n (100.0%)	V5n / V5s (100.0%)
Outside zone1 in person_obstructed	100.00% ± 0.00	100.00% ± 0.00	All models (100.0%)	All models (100.0%)

For the zone-focused variants, cloud inference again achieved higher mean recall than edge inference. For the no-zone-focused variants, both deployment modes remained comparatively strong, and the outside-zone condition reached perfect recall across all models. In the combined person-obstruction total, precision was high for both deployments: 97.8–99.3% on edge and 98.9–99.6% in cloud mode. The larger variation was in missed occupied states rather than in false-positive over-counting. For projected lighting control, this means that the main risk in this scenario is an occupied zone remaining inactive, rather than widespread unnecessary activation.

6.1.4 Model choice across scenarios

The scenario results show that no single YOLOv5 variant is best in every condition. On edge, the best model changes between the evaluated scenarios: V5x is strongest in the daytime no-obstruction setting and behind-object-obstruction phase, V5s-kd is strongest in the night-time no-obstruction setting, V5l-kd is strongest outside the object obstruction, and V5n is strongest in several person-obstruction variants. The cloud results also vary by scenario, with V5l/V5x strongest in the daytime

no-obstruction case, V5s strongest at night and behind the object obstruction, V5l-kd and several non-distilled models reaching perfect recall outside the object obstruction, and V5m strongest in several zone-focused person-obstruction variants. Figure 6.1 visualises the same model-dependence separately for edge and cloud execution.



(a) Edge **(b)** Cloud

Figure 6.1: Cross-scenario performance heatmaps. Darker cells indicate better performance, measured as 100−missed percentage. The figure shows that model performance is condition-dependent rather than dominated by a single model across all scenarios.

The controlled replay scenarios provide two main RQ1 findings. First, cloud inference generally produced stronger recall, especially under low light and obstruction. Second, model configuration still mattered inside each placement, so placement and model size cannot be treated as independent decisions.

6.2 Decision-path latency and runtime burden

This section reports the timing and runtime-burden results that support RQ1 and RQ2. The full-day timing table is derived from the full-day stress-trace logging outputs used for the later lighting and power analyses, but it aggregates the available timing fields separately from the energy calculations. The edge full-day timing fields report local model inference and local request time from the controller logs. The cloud timing fields report model inference together with cloud total time and edge-observed transport time from cloud telemetry. For the cloud values, transport time is the broadest timing boundary and is therefore the closest available indicator of the cloud decision path.

The edge timing results show a large model-size effect. Mean local request time ranged from 63.6 ms for V5n to 627.2 ms for V5x. Cloud inference reduced the model-inference part of the timing substantially for all larger models: V5x averaged 53.9 ms for cloud inference compared with 594.7 ms for edge inference. However, the cloud transport boundary was higher than the fastest edge local decision path, ranging from 117.9 ms to 142.1 ms across models. The latency result is not that one placement is always faster. Cloud placement reduced heavy-model inference time, while edge placement avoided transport overhead and gave the smallest model the lowest observed end-to-end timing.

Table 6.5: Full-day decision-path timing by model. Values are model-level means in milliseconds.

Model	Edge infer	Edge request	Cloud infer	Cloud total	Cloud transport
V5n	35.6	63.6	10.1	28.8	118.7
V5s-kd	78.6	108.2	10.0	30.5	117.9
V5s	76.3	105.7	10.3	31.0	120.0
V5m	190.5	222.2	17.3	38.5	126.2
V5l-kd	345.1	377.4	29.8	51.7	131.4
V5l	356.3	387.4	29.6	50.9	131.5
V5x	594.7	627.2	53.9	75.0	142.1

6.2.1 Cloud in-flight tuning

The cloud in-flight tuning results complement the full-day timing analysis by reporting how request-concurrency settings affect cloud-runtime behaviour. The tuning run uses the day no-obstruction cloud setup as a stable visual input for isolating request scheduling effects; it should not be read as a complete representation of the later full-day stress trace. Table 6.6 reports total drop, queue drop, stale-result percentage, and total cloud time for each tested configuration.

Table 6.6: Cloud in-flight tuning results for the day no-obstruction cloud setup.

Cfg	Runs	Total drop %	Queue drop %	Stale %	Cloud total ms
1	15	46.92 ± 3.19	46.81 ± 3.19	0.00 ± 0.00	28.91 ± 16.02
2	15	8.16 ± 7.79	7.08 ± 7.69	8.71 ± 3.92	37.44 ± 25.54
3	15	2.18 ± 0.94	0.00 ± 0.00	6.00 ± 1.60	39.38 ± 29.93
4	15	3.11 ± 1.06	0.00 ± 0.00	6.33 ± 1.79	37.08 ± 26.40
8	15	4.52 ± 1.19	0.00 ± 0.00	6.48 ± 2.54	38.31 ± 30.47

The in-flight tuning results show a trade-off between queue dropping and stale results. With only one in-flight request, queue drops were high at 46.81% on average. Increasing the setting to three or more removed queue drops in this test, but stale-result behaviour remained between 6.00% and 6.48%. These counters are response-quality issues, not only runtime diagnostics. A queue drop means that a frame never contributes to a lighting decision, while a stale result means that a decision may arrive after the observed scene has already changed.

6.2.2 Full-day cloud runtime counters

The full-day cloud run provides the runtime-burden evidence for continuous perception with remote inference. Table 6.7 summarises the cloud-specific counters across models. The edge full-day run does not include corresponding queue counters because edge inference does not send frames to a remote inference queue.

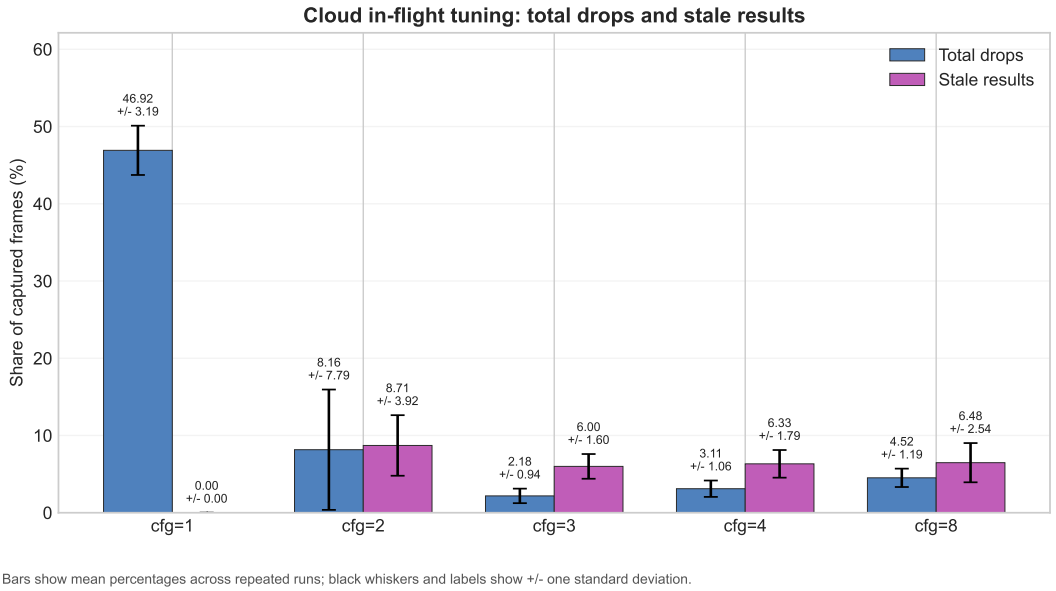


Figure 6.2: Total-drop and stale-result percentages by maximum in-flight configuration in the cloud tuning run. Bars show the mean across repeated runs, while black whiskers and labels show ±1 standard deviation.

Table 6.7: Cloud full-day runtime counters across models. Values are ranges across the seven evaluated models.

Counter	Range	Lowest model	Highest model
Accepted remote requests	100,618–113,587	V5l-kd	V5l
Latest-slot drops	104,334–122,757	V5n	V5l-kd
Cloud queue drops	206–4,743	V5n	V5x
Stale results	5,985–9,269	V5l-kd	V5s
Failures	0	All models	All models
In-flight high watermark	4	All models	All models

The cloud full-day run shows that the cloud path remained operational in the sense that no request failures were logged, but it also shows substantial backpressure and stale-result behaviour. Because Table 6.7 is a range summary, it identifies the lowest and highest model per counter but does not show every per-model combination of stale counts, queue drops, timing, and energy. These values limit how strongly the cloud results can be interpreted as stable control-loop behaviour. They support communication-sensitive claims only for the observed queue, stale-result, and transport telemetry, not for untested packet loss, outage, or degraded-network conditions.

6.3 Projected lighting behaviour and energy

This section reports the full-day analytical lighting results used for RQ3 and the processing-energy part of RQ2. The analysed window is 19,999.267 s for all cloud runs and for all edge runs except edge V5n, which is 19,999.733 s. This corresponds to approximately 5.56 h and is reported as a common full-day trace window in the aggregate tables. Lighting energy is projected from reconstructed zone-state intervals and nominal fixture wattages. Processing energy is derived from measured edge total power for edge runs and from combined measured edge-side plus logged cloud GPU power for cloud runs.

6.3.1 Projected lighting behaviour over time

The full-day trace was reconstructed into zone-aware, motion-detector, and physical-switch lighting intervals. The physical-switch baseline is active for the full analysed duration. The motion-detector baseline is also close to continuous activation in this stress trace because the eight-minute hold time keeps all lamp-bearing zones active after detected activity. The zone-aware controller instead activates the red, yellow, and blue zones independently, which reduces projected lighting energy when occupancy is spatially uneven.

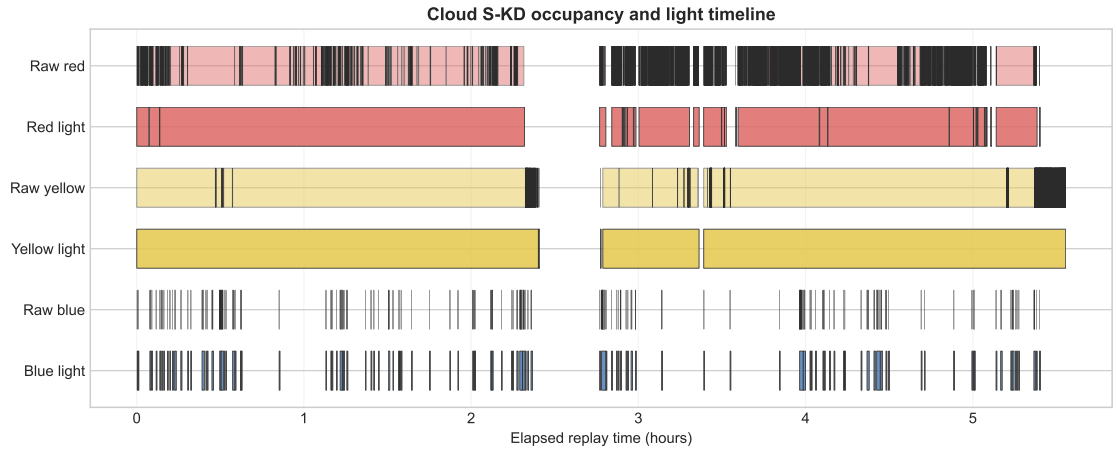


Figure 6.3: Representative projected lighting timeline for the cloud V5s-kd full-day run. The timeline compares raw reconstructed occupancy and projected zone-aware activation for the red, yellow, and blue lamp-bearing zones.

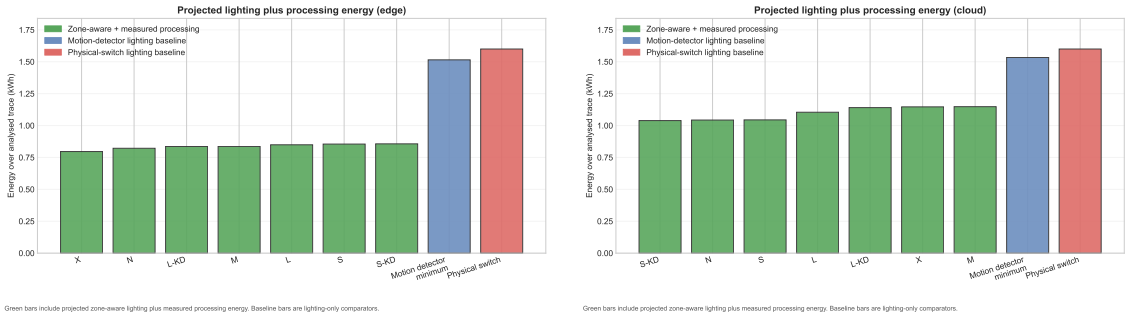
False-positive lighting time was limited but model-dependent. In the edge full-day run, false-positive lighting time ranged from 0.0 s to 78.3 s. In the cloud full-day run, it ranged from 0.0 s to 272.7 s. The false-positive intervals are small compared with the full analysed duration, but they are still relevant for projected lighting behaviour because they represent analytical lighting activity caused by detections outside the lamp-bearing zones.

6.3.2 Projected lighting energy

Table 6.8 summarises projected lighting energy before processing energy is added. Figure 6.4 then shows the application-level comparison in which processing energy is added to the zone-aware model bars while the baselines remain lighting-only analytical comparators. This difference in boundary is deliberate and is used to show both the lighting-control effect and the processing-energy burden. The zone-aware controller reduced projected lighting energy relative to both baselines for every evaluated model and placement.

Table 6.8: Projected lighting energy over the full-day trace. Ranges are across the seven evaluated models.

Target	Zone-aware kWh	Motion kWh	Switch kWh	Saving vs motion	Saving vs switch
Edge	0.762–0.825	1.515–1.522	1.600	45.69–49.69%	48.44–52.36%
Cloud	0.815–0.907	1.534–1.600	1.600	43.29–48.25%	43.30–49.06%



(a) Edge **(b)** Cloud

Figure 6.4: Projected full-day energy by model and analytical control mode. The zone-aware model bars include projected lighting energy plus measured processing energy, while the baseline bars are lighting-only analytical comparators; the figure should therefore be read as a bounded application-level comparison rather than as a pure lighting-energy chart.

The edge V5x run produced the lowest lighting-only projected zone-aware energy, at 0.762 kWh, and the highest projected savings in the edge full-day run before processing energy was added. In the cloud full-day run, V5s-kd produced the lowest

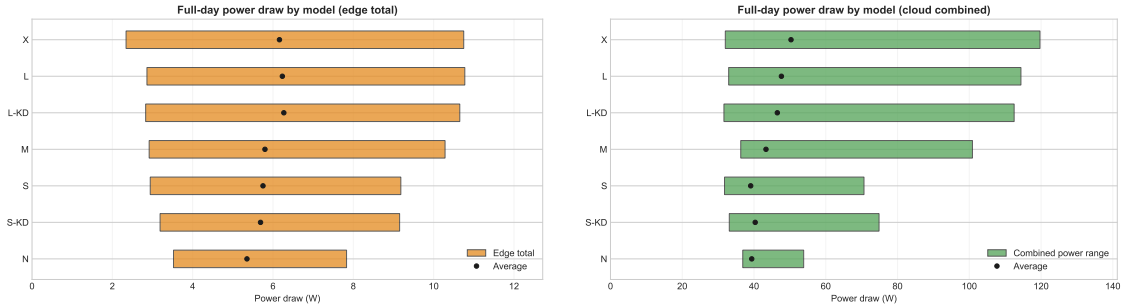
lighting-only projected zone-aware energy, at 0.815 kWh. These differences reflect the fact that the lighting projection depends on the reconstructed occupancy signal, not only on the nominal control policy.

6.3.3 Processing power and operational energy

Table 6.9 separates processing power and processing energy from projected lighting energy. This separation is needed because a controller that saves lighting energy can still be unattractive if continuous perception consumes too much power.

Table 6.9: Processing power, processing energy, and net-energy savings relative to the analytical baselines over the full-day trace. Ranges are across models.

Target	Average power	Processing kWh	Net saving vs motion	Net saving vs switch	Boundary
Edge	5.35–6.27 W	0.0297–0.0348	0.660–0.719 kWh	0.743–0.804 kWh	Edge total power
Cloud	39.08–50.31 W	0.2171–0.2795	0.452–0.550 kWh	0.452–0.561 kWh	Edge total + cloud GPU



(a) Edge **(b)** Cloud

Figure 6.5: Measured processing-power ranges by model. Edge values use Jetson total power; cloud values combine logged cloud GPU power with measured edge-side total power.

The edge runs required substantially less processing energy than the cloud runs under the applied measurement boundary. Edge processing energy ranged from 0.0297 kWh to 0.0348 kWh, while cloud processing energy ranged from 0.2171 kWh to 0.2795 kWh. The difference is mainly caused by the logged cloud GPU component, which ranged from 0.1884 kWh to 0.2512 kWh, while the paired cloud-mode edge-side component ranged from 0.0282 kWh to 0.0302 kWh. The cloud edge-side contribution alone was therefore similar to the edge-side power level, but the combined cloud-GPU-plus-edge value was much higher.

6.3.4 Net energy results

For each baseline, net saving is calculated as baseline projected lighting energy minus the sum of zone-aware projected lighting energy and processing energy. Under that definition, all evaluated model-placement combinations still produced positive net savings relative to both analytical baselines. The magnitude differed by placement. Edge net savings were 0.660–0.719 kWh relative to the motion-detector baseline and 0.743–0.804 kWh relative to the physical-switch baseline. Cloud net savings were lower, at 0.452–0.550 kWh relative to the motion-detector baseline and 0.452–0.561 kWh relative to the physical-switch baseline.

Expressed as percentages of the corresponding baseline lighting energy, edge net savings were 43.58–47.43% relative to the motion-detector baseline and 46.46–50.22% relative to the physical-switch baseline. Cloud net savings were 28.24–34.53% relative to the motion-detector baseline and 28.25–35.06% relative to the physical-switch baseline. These results indicate that the zone-aware analytical controller can remain energy-positive even after processing energy is included, but they also show that the cloud placement has a higher processing-energy burden under the reported boundary.

6.4 Baseline automation comparison results

The baseline comparison answers the application-level part of RQ3. The physical-switch baseline represents keeping all evaluated fixtures active for the full analysed period, which resulted in approximately 1.600 kWh of projected lighting energy. The motion-detector baseline was slightly lower but close to the physical-switch baseline, at 1.515–1.522 kWh in the edge reconstruction and 1.534–1.600 kWh in the cloud reconstruction. The zone-aware controller was lower for all models, at 0.762–0.825 kWh for edge and 0.815–0.907 kWh for cloud.

The baseline result supports the value of spatially resolved zone control in this trace. The advantage does not come from detecting occupancy in general, because the motion-detector baseline also reacts to occupancy. It comes from using the reconstructed zone state to avoid lighting all lamp-bearing zones when only part of the space is projected as occupied. The result is analytical rather than directly actuated, but it connects the perception output to the projected energy and control-behaviour consequences that matter for lighting control.

6.5 Summary by research question

For RQ1, the results show that inference placement and model configuration both affected zone-occupancy performance and latency. Cloud inference generally produced higher recall, especially under night-time and obstruction conditions, and it reduced heavy-model inference time. Edge inference avoided cloud transport overhead and produced the lowest observed decision-path timing for the smallest model.

For RQ2, the results show that runtime burden differs by placement. Edge processing had low measured power and no remote queue behaviour. Cloud processing introduced transport, queue, and stale-result behaviour, and the measured cloud-GPU-plus-edge power was much higher than edge-only processing power. No cloud request failures were logged in the full-day run, but queue drops and stale results remained relevant runtime-burden indicators.

For RQ3, the projected zone-aware controller reduced lighting energy relative to both the motion-detector and physical-switch baselines across all evaluated models. Net savings remained positive after processing energy was included, but the net-energy margin was larger for edge deployment than for cloud deployment under the reported measurement boundary. These findings support the claim that zone-aware perception can improve projected lighting energy outcomes, while also showing that the placement choice changes the balance between perceptual robustness, timing, runtime burden, and processing energy.

7

Discussion

This chapter interprets the results in relation to the research questions, the theoretical framework, and the practical deployment setting. The purpose is not to repeat every result from Chapter 6, but to explain what the results mean for a stereo-depth-assisted, zone-based perception-and-decision pipeline whose inference can run either locally on constrained edge hardware or remotely through a cloud service. The discussion separates empirical evidence from replayed occupancy and runtime experiments, analytical evidence from reconstructed lighting and energy models, and design implications that follow from the combination of those findings.

The central finding is that inference placement should be treated as an architectural design decision rather than as a purely technical implementation detail. Cloud inference generally improved zone-occupancy recall in the more difficult perception scenarios, particularly under low light and obstruction. Edge inference, however, avoided cloud transport overhead, required substantially less measured processing energy, and reduced privacy exposure by keeping the inference path local. The results do not identify one universally superior deployment strategy, and they should not be read as evidence of long-term operational reliability. They show that deployment choice changes the balance between perceptual robustness, decision-path latency, runtime burden, energy consumption, and privacy risk within the evaluated short-duration traces.

7.1 Interpreting RQ1: Placement, model choice, and control-relevant perception

RQ1 asks how inference placement and model configuration affect zone-occupancy accuracy, robustness, and end-to-end decision-path latency. The results indicate that both factors matter, and that they interact with scene difficulty. Cloud inference generally produced stronger recall than edge inference, especially in the night-time and obstruction scenarios. This supports the expectation from the theoretical framework that constrained local hardware can become limiting when the perception task is difficult. The stronger cloud results in those conditions suggest that remote compute can make the detector part of the pipeline more robust when illumination, occlusion, or multiple visible people increase the difficulty of the visual

task.

The edge results also show that local inference should not be dismissed as inadequate. In the daytime no-obstruction scenario, the best edge models were close to the cloud models, and in the no-zone parts of the person-obstruction scenario both placements performed strongly. Practical lighting deployments do not necessarily spend all of their time in the most difficult visual state. In a stable room with good illumination, a well-selected edge model may provide sufficient control-relevant occupancy quality while avoiding communication dependence. The practical implication is that the deployment decision should be made with reference to expected scene conditions, not only to an aggregate accuracy number.

The robustness findings also clarify which types of errors are most relevant for lighting control. Missed occupied states are more problematic than ordinary detector misses in an abstract object-detection benchmark because a missed occupied zone may leave the corresponding lights inactive. The night-time no-obstruction result is not merely a detector-quality issue; it is a control-quality issue. A system that appears reliable in daytime may become unacceptable if the same room is used under darker conditions and the controller fails to reconstruct occupied zones. Similarly, the behind-obstruction phase in the object-obstruction scenario shows why partial occlusion is relevant in ordinary rooms. Furniture, standing people, open doors, and other objects can hide parts of a body. A lighting controller that only works when occupants are fully visible would be fragile in realistic use.

The person-obstruction scenario adds a different interpretation. The results show that multi-person and partially overlapping detections can affect recall even when precision remains high. This suggests that the main risk in this study is not primarily that the controller invents large amounts of occupancy, but that it sometimes fails to reconstruct all relevant occupied states. For lighting, this creates an asymmetric design concern. False positives waste energy and can reduce perceived appropriateness, but false negatives can produce a more direct user-facing failure by leaving an occupied area insufficiently lit. This asymmetry supports the choice in Chapter 6 to emphasise recall and missed occupancy as central measures for RQ1.

7.1.1 Model configuration as a deployment variable

The model comparison shows that no single YOLOv5 variant dominated all scenarios, and that the best choice depended on both scene conditions and inference placement. This means model selection cannot be separated from deployment design: a larger or more capable detector may improve recall in one setting but become less attractive if it increases local latency, processing energy, or cloud stale-result behaviour. This aligns with prior work on efficient AI and embedded detection [4, 23]. For this thesis, the relevant unit of evaluation is therefore the deployed controller as a whole: whether the chosen model and placement produce useful and timely zone occupancy under the intended constraints.

7.1.2 Latency as part of response quality

The decision-path latency results add nuance to the accuracy findings. Cloud inference reduced the model-inference component for the heavier models, but it also introduced transport overhead. Edge inference avoided that transport overhead, and the smallest edge model produced the lowest observed decision-path timing. This means that the latency answer to RQ1 is conditional rather than absolute. Cloud placement can make heavy models much faster at the inference step, but the full cloud decision path is still shaped by communication. Edge placement can be faster for small local models because it keeps the loop local, but local latency grows substantially as model size increases.

For occupant-facing automation, projected lighting response quality is not determined only by detector correctness. It also depends on whether occupancy decisions arrive while the scene state is still relevant. The results support the theoretical claim in Chapter 4 that latency is an application-level property of the control loop, not only a benchmark property of the detector. A cloud model that is accurate but frequently delayed can produce stale projected control actions. Conversely, an edge model that is fast but misses difficult occupied states can produce timely but incorrect analytical actions. The system designer must therefore balance correctness and timeliness rather than maximising either one in isolation.

7.2 Interpreting RQ2: Runtime burden and operational energy

RQ2 asks how inference placement and model configuration affect runtime burden and operational energy consumption during continuous perception, including communication overhead where applicable. The results show that runtime burden differs qualitatively between edge and cloud placement. Edge inference is constrained by local compute and power, but it does not use the remote request queue and therefore does not produce cloud-specific queue drops, stale results, or transport telemetry. Cloud inference shifts the heaviest detector computation away from the local device, but introduces request scheduling, transport delay, queueing, stale-result behaviour, and remote GPU power consumption.

The cloud in-flight tuning results show that cloud operation cannot be evaluated only by running a model remotely and measuring inference time. The maximum number of in-flight requests changed queue-drop behaviour substantially. Very low concurrency caused large queue losses, while higher settings removed queue drops in the tuning experiment but still left stale-result behaviour. This means that cloud deployment is an operational service configuration problem, not only a model-deployment problem. The cloud endpoint, request policy, queue policy, and stale-result handling all become part of the effective controller.

From a control perspective, queue drops and stale results should be treated as response-quality signals. A queue drop means that a captured frame never con-

tributes to the lighting decision path. A stale result means that the detector output may describe an earlier state of the room. Both can affect whether the reconstructed occupancy state matches the current physical situation. This is why the cloud runtime counters are not merely diagnostic implementation details. They provide evidence about how communication-sensitive architecture affects the control signal that would later drive lighting.

The full-day cloud counters also show that the cloud setup remained operational in the narrow sense that request failures were not logged, while still producing substantial backpressure and stale-result behaviour. This distinction matters for interpreting robustness. The results support claims about observed transport, queue, and stale-result behaviour in the implemented configuration. They do not support stronger claims about packet loss, outages, jitter tolerance, or degraded-network robustness, because no controlled degraded-network experiment was conducted. That boundary should remain explicit in the thesis, especially because network robustness is often a decisive practical factor in cloud-dependent automation.

7.2.1 Operational energy and measurement boundaries

The energy results show a clear processing-energy difference between edge and cloud placement under the reported measurement boundary. Edge processing consumed much less measured processing energy than cloud processing. The cloud runs included edge-side power plus logged cloud GPU power, and the cloud GPU contribution made the combined processing-energy term much higher than the edge-only processing term. The result supports the conclusion that cloud placement can improve difficult perception cases, but it does not do so without an energy cost.

This finding should be interpreted within the stated measurement boundary. The edge processing-energy result is based on Jetson total-power telemetry. The cloud processing-energy result includes logged cloud GPU power and measured edge-side power, but excludes full host power, CPU, memory, storage, network equipment, and datacentre overhead. The thesis can state that cloud processing had a higher measured edge-plus-cloud-GPU energy burden in the evaluated setup. It cannot claim to have measured complete cloud infrastructure energy from wall plug to datacentre overhead.

The exclusion of full cloud-host and infrastructure power also means that the cloud processing-energy estimate is likely conservative in relation to complete infrastructure energy, although the exact direction and magnitude cannot be quantified from the present data. Because the unmeasured components are outside the measurement boundary, they should not be added through unsupported assumptions. The better academic interpretation is to report the boundary transparently and treat the comparison as a bounded deployment-energy comparison. This is consistent with the thesis's broader aim: to understand architectural trade-offs in one implemented system, not to produce a complete lifecycle assessment of cloud computing.

The energy results also show why the net-energy perspective is necessary. A con-

troller can save lighting energy while still consuming energy for sensing, inference, communication, and logging. If that processing energy is large enough, it can reduce or eliminate the benefit of more selective lighting control. In this study, net savings remained positive for both placements relative to the analytical baselines, but the margin was larger for edge deployment. This suggests that the zone-aware controller can be energy-positive in the evaluated trace, while also showing that deployment placement changes how much of the lighting saving is consumed by the perception infrastructure.

7.3 Interpreting RQ3: Projected controller behaviour and baseline comparison

RQ3 asks how the proposed controller variants affect projected lighting behaviour and projected net-energy outcomes relative to baseline automation strategies. The results indicate that the zone-aware controller reduced projected lighting energy relative to both the motion-detector and physical-switch baselines across all evaluated model-placement combinations. This supports the practical value of spatially resolved occupancy. The benefit does not come simply from knowing whether someone is present in the room. The motion-detector baseline also reacts to occupancy. The benefit comes from using zone occupancy to avoid lighting all lamp-bearing areas when only part of the monitored space is occupied.

The physical-switch baseline represents the least adaptive case: all mapped fixtures are treated as active for the full analysed period. It is a useful reference because it shows the energy consequence of no automated adaptation. The motion-detector baseline is more realistic as a conventional automation comparator, but in the analysed stress trace it remained close to continuous activation because the hold time kept the lamp-bearing zones active after detected activity. The zone-aware controller differs by controlling the red, yellow, and blue zones independently. This finer control granularity is what produces the projected lighting-energy reduction.

The response-quality interpretation is more cautious. Because the evaluated runtime ends at occupancy decisions and logging, fixture switching was not measured physically. The thesis therefore cannot claim that actual light fixtures would switch exactly as projected, that occupants would perceive the transitions as comfortable, or that the controller would avoid all visible oscillation in deployment. What the thesis can claim is that, when the logged occupancy states are reconstructed through the defined analytical control model, the zone-aware strategy produces less projected lighting activation than the two baselines while maintaining positive net-energy savings after processing energy is included.

False-positive lighting time was limited but model-dependent. This supports a more specific interpretation of projected lighting behaviour. The zone-aware controller's energy benefit depends not only on whether it detects occupied zones, but also on whether it avoids unnecessary activation outside lamp-bearing zones. False-positive intervals are small relative to the analysed window, but they still matter because they

represent projected control actions that would be visible to occupants or building operators if the system were connected to fixtures. In a real deployment, even short erroneous activations could reduce trust if they occur at noticeable moments. The present results should therefore be interpreted through both accumulated energy and the temporal character of erroneous activations.

7.3.1 Role of projected lighting analysis

Although the actuation result is analytical, it is still useful because it links the perception pipeline to the application-level question. A detector benchmark alone would not show whether improved perception translates into lower lighting energy or better projected control behaviour. By reconstructing lighting intervals from zone states, the thesis shows how perception errors, model choice, and placement choice can influence the behaviour of a lighting policy.

The boundary is that lighting energy is calculated from nominal fixture wattages, reconstructed zone intervals, and deterministic hold rules rather than measured from physical fixture actuation. The model also treats the fixtures as binary on/off states and does not include dimming levels, daylight contribution, illuminance feedback, or occupant comfort preferences. The appropriate interpretation is therefore that the zone-aware controller shows application-level potential, not that it has already been proven as a finished smart-lighting product.

7.3.2 Selective offloading as a bounded design implication

The combined results support selective offloading as a design implication, but not as a tested result. Edge inference was energy-efficient, local, and fast for smaller models, while cloud inference improved perception in more difficult scenes. This suggests a future design hypothesis: a practical controller could treat edge inference as the default and reserve cloud inference for conditions where the local signal is likely to be unreliable. Possible triggers include low confidence, unstable depth near zone boundaries, low illumination, overlapping occupants, or rapid occupancy-state changes, but these triggers were not implemented or validated in this thesis.

This implication must remain bounded by the evidence. The thesis did not implement or evaluate a hybrid controller, and selective cloud use would introduce additional complexity, failure modes, and privacy questions. A system that sometimes sends frames to the cloud still needs a data-protection justification for those transmissions, as well as clear fallback behaviour when cloud inference is unavailable or too slow. Selective offloading is therefore a deployment pattern suggested by the observed trade-off between edge efficiency and cloud robustness, not a conclusion validated empirically in this study.

7.4 Relation to theory and prior work

The findings align with the theoretical framework in several ways. First, they support the distinction between person detection and zone occupancy. The relevant output in this thesis is not simply whether a person is detected in the image, but whether the resulting spatial state is useful for lighting control. The controlled scenarios show that scene difficulty can affect this derived occupancy state even when the detector family is held constant. This supports the thesis's decision to evaluate the pipeline at the control-signal level rather than only at the detector level.

Second, the results reinforce the idea that stereo-depth-supported spatial assignment is valuable but not sufficient by itself. Depth information enables mapping detections to zones, but it does not eliminate detector misses, occlusion problems, lighting-condition effects, or timing problems. The system therefore depends on both the detector and the spatial assignment stage. This is consistent with the theory chapter's claim that depth is an enabling mechanism rather than a guarantee of correct localisation.

Third, the results support the related-work argument that efficient AI and embedded detection studies need to be connected to application-level consequences. Prior work on embedded object detection often reports throughput, power, and model performance [4]. Those measures are necessary, but this thesis shows why they are not sufficient for a lighting-control application. A model's practical value depends on how its errors and timing behaviour propagate into zone occupancy, projected lighting actions, and net energy. This is the thesis's main extension beyond detector-level comparison.

Fourth, the cloud results are consistent with the broader inference-placement literature: remote execution can shift compute demand away from the client but introduces communication-dependent behaviour [9]. In this study, that behaviour appears as transport time, queue drops, and stale results. The result is especially relevant for building automation because the control output is visible in the physical environment. A delayed web response and a delayed light response are not perceived in the same way, but both illustrate that system latency becomes a user-facing quality attribute.

Finally, the findings support the privacy-aware architectural reasoning in Chapters 2 and 4. Cloud inference may improve difficult perception scenarios, but it also increases data-transfer exposure. Edge inference may be weaker in some difficult scenarios, but it better supports data minimisation because raw or derived visual information can remain local. This means that privacy is not a separate constraint added after the technical optimisation. It is part of the same architectural trade-off.

7.5 Practical implications and edge/cloud trade-offs

For practice, the main implication is that inference placement and model configuration should be evaluated through their application-level consequences: zone-occupancy quality, timing, runtime burden, projected lighting behaviour, net energy, and privacy exposure.

Deployment decisions should be scenario-aware. If a room is consistently well-lit, has limited occlusion, and has low privacy tolerance for data transfer, edge inference may be the more appropriate default. If the room has frequent obstruction, poor lighting, or complex multi-person scenes, cloud inference or a hybrid design may provide better perception quality. This recommendation is not universal. It depends on the performance threshold required by the lighting application and on how much communication dependence and data transfer are acceptable.

Observability should also be designed into the controller from the beginning. The value of the analysis depended on logs for zone states, timing, queue drops, stale results, and power. Without those logs, the system would be difficult to evaluate beyond anecdotal impressions. For deployable AI control systems, observability is not only useful during development. It is a requirement for maintenance, fault diagnosis, privacy auditing, and continued performance assessment.

Cloud inference should be treated as an operational subsystem. It requires request scheduling, queue management, stale-result handling, timeout behaviour, and failure policies. A cloud detector that performs well under isolated testing may still behave poorly if the surrounding request pipeline drops too many frames or returns results too late. A deployable implementation therefore needs to specify not only the model and endpoint, but also the operational policy that determines which frames are sent, how late results are handled, and what fallback is used when the cloud path is degraded.

Energy evaluation should include both application energy and processing energy. A lighting controller can reduce fixture energy while increasing sensing and compute energy. The net-energy formulation used in the thesis is therefore a useful design pattern for similar systems. It makes the trade-off visible and prevents the analysis from treating the perception infrastructure as free. This is especially important for AI-enabled building systems that run continuously.

Finally, privacy-aware architecture should be considered at the same time as accuracy and latency. Edge processing, cloud processing, and hybrid processing expose data differently. The thesis does not claim that cloud processing is inherently unacceptable, but it shows that the benefits of cloud inference must be weighed against transmission, retention, and legal-context considerations. A practical design process should therefore include data minimisation, retention limits, access control, and transparency mechanisms as first-order requirements.

7.6 Limitations and validity

The results should be interpreted within the scope of the study. The evaluation is based on one system architecture, one camera type, one local edge platform, one cloud setup, and a limited set of scenarios. These choices make the study concrete and traceable, but they also limit generalisability. A different room layout, camera angle, lighting condition, network path, detector family, edge device, or fixture arrangement could change the balance between edge and cloud deployment.

7.6.1 Internal validity

Internal validity concerns whether the observed differences can reasonably be attributed to the studied factors. The replay-based methodology strengthens internal validity because each model-placement comparison can be evaluated on the same visual input within fixed analysis windows. This reduces the risk that differences are caused by different occupant movements or different scene sequences. Repeated runs also reduce reliance on single-run behaviour.

However, internal validity is still affected by calibration, logging, and scenario-definition choices. Zone assignment depends on camera calibration, depth quality, and the mapping between detections and physical zones. If the calibration is slightly wrong, the controller may misclassify boundary cases independently of detector quality. The manually defined expected traces and scenario windows also introduce researcher judgement. These choices are necessary for frame-level scoring, but they should be recognised as possible sources of bias. The thesis mitigates this through fixed windows and consistent scoring, but it does not remove the need for careful interpretation.

7.6.2 Construct validity

Construct validity concerns whether the measures actually represent the concepts being discussed. The thesis uses recall, precision, missed occupancy, latency, queue drops, stale results, power, and projected lighting energy to operationalise accuracy, robustness, projected lighting behaviour, runtime burden, and net energy. These measures are appropriate for the implemented system because the runtime output is zone occupancy and logging, not physical lighting actuation.

The main construct-validity limitation is that response quality is only partly measured directly. Decision-path timing and stale results are measured indicators, and reconstructed lighting intervals are analytical indicators, but together they do not fully capture occupant-perceived lighting quality. Actual comfort depends on illuminance levels, transition smoothness, glare, visual tasks, occupant expectations, and tolerance for occasional errors. Because physical dimming and occupant feedback are outside the implemented evaluation, response quality should be interpreted as projected control behaviour rather than measured user experience.

Another construct-validity boundary concerns energy. Lighting energy is projected

analytically from nominal fixture wattages and reconstructed activation intervals. Processing energy is measured or logged within the stated boundaries. These are useful and transparent constructs, but they are not the same as whole-building energy measurement. The thesis therefore supports conclusions about projected lighting and bounded processing energy, not about total building-level energy performance.

7.6.3 External validity

External validity concerns transferability. The findings are most directly applicable to camera-based, depth-assisted, zone-aware lighting controllers with similar hardware constraints and similar control logic. They should transfer cautiously to other rooms, other fixture layouts, other sensing technologies, or other building-automation tasks. For example, a room with stronger daylight variation, more reflective surfaces, different furniture, lower ceilings, or more frequent crowding could change detection quality and depth reliability. A different edge device could also make larger local models practical, reducing the need for cloud inference.

The study is also limited by duration. The full-day trace gives a more realistic temporal sequence than short controlled trials, but it does not establish long-term reliability. Building systems must operate over months or years. Over that time, camera alignment can change, lighting conditions can vary seasonally, furniture can move, models can become outdated, and maintenance practices can affect performance. Long-term deployment remains necessary before the results can be generalised as operational evidence for real building management.

7.6.4 Reliability and reproducibility

Reliability is supported by the log-based evaluation pipeline, repeated controlled runs, fixed scenario windows, and explicit metric definitions. These choices make the analysis more repeatable than an informal prototype demonstration. They also make it possible to trace results from controller logs to reconstructed occupancy and projected lighting outcomes.

At the same time, reproducibility depends on the availability of the same software, model files, hardware, camera calibration, input traces, and analysis scripts. Small changes in any of these elements could affect the results. This is typical for applied systems research, and it means that any repetition or extension of the study depends on careful documentation of configuration files, software versions, camera placement, zone definitions, and preprocessing steps.

7.7 Ethical and privacy implications

The ethical issue most directly connected to the results is the trade-off between performance and privacy exposure. Cloud inference improved several difficult perception cases, but it also requires transmitting image data or derived visual information away from the local device. Edge inference reduces that exposure but may provide

lower recall in difficult conditions. The thesis therefore highlights a practical ethical tension: the configuration that performs best technically is not automatically the configuration that is most proportionate from a privacy perspective.

This matters because the system is camera-based and intended for indoor environments where occupants may be employees, students, visitors, or other people who have limited practical ability to avoid the monitored space. Even if the system does not perform face recognition or long-term identity tracking, raw images and person-related metadata can still be sensitive. The privacy discussion should therefore remain concrete: what is collected, what is transmitted, what is stored, who can access it, how long it is retained, and whether the same control goal could be achieved with less intrusive data.

The results also support the importance of transparency. Occupants should be able to understand that the system uses camera-based sensing for lighting control, not general surveillance. In a practical deployment, visible notices, clear documentation, data-retention policies, and accountable system ownership are necessary. These are not peripheral concerns. They affect whether the system can be responsibly deployed even if the technical performance is acceptable.

There is also an ethical dimension to reliability. A lighting controller that fails under low light or obstruction can affect comfort and accessibility. If users rely on automated lighting, missed occupancy could create unsafe or inconvenient conditions. This is another reason why recall and robustness are important. Energy saving should not be pursued in a way that makes the environment less usable for occupants. Within the present study, this means that the positive net-energy result should be read together with the observed perception limitations rather than as a stand-alone justification for deployment.

7.8 Overall interpretation

Taken together, the results support a cautious but useful conclusion. Zone-aware, stereo-depth-assisted perception can improve projected lighting outcomes compared with less spatially specific baselines, and the net-energy result remains positive after processing energy is included. However, the architecture used to produce the occupancy signal changes the trade-off. Cloud inference improves difficult perception cases but introduces communication-sensitive runtime behaviour, higher measured processing energy under the reported boundary, and greater privacy exposure. Edge inference is more energy-efficient and privacy-preserving in the evaluated setup, but it is more vulnerable to difficult visual conditions and heavier local models.

The thesis therefore does not argue that smart lighting should always be edge-based or always be cloud-based. Its more defensible contribution is to show how the choice should be reasoned about. A deployable controller must be evaluated as a control system, not only as a detector. It must be judged by whether it produces sufficiently correct, timely, stable, energy-positive, and privacy-aware occupancy

decisions under the conditions in which it will actually be used. Within the scope of this thesis, treating inference placement and model configuration as architectural variables makes those trade-offs visible and gives the results a clearer software-engineering contribution than a detector benchmark alone would provide.

8

Conclusion

This thesis investigated how inference placement and model configuration affect a stereo-depth-assisted, zone-based perception-and-decision pipeline for lighting control. The work was motivated by a practical software-engineering problem: camera-based lighting control is not only a question of whether people can be detected, but whether the resulting zone-occupancy signal can be produced accurately, promptly, continuously, and within acceptable energy and privacy boundaries. To study this, the thesis implemented a shared perception-and-decision pipeline in which person detections are combined with stereo-depth-supported spatial assignment, and compared edge and cloud inference variants across controlled replay scenarios and full-day analytical lighting reconstructions.

The overall conclusion is that inference placement should be treated as a central architectural design decision in AI-enabled building automation. The results do not identify edge or cloud inference as universally superior. Instead, they show that placement changes the balance between perception quality, latency, runtime burden, energy consumption, observability, and privacy exposure. Model configuration also matters, but it does not resolve the deployment question by itself, since no single YOLOv5 variant was best across all evaluated scenarios.

8.1 Answers to the research questions

For RQ1, the results show that both inference placement and model configuration affected zone-occupancy accuracy, robustness, and decision-path latency. Cloud inference produced stronger recall in the more difficult scenarios, especially under low light and obstruction. This was visible in the night no-obstruction scenario, where mean recall was substantially higher for cloud than for edge, and in the behind-object-obstruction phase, where the edge results degraded more strongly. Edge inference remained competitive in less demanding conditions, particularly in the daytime no-obstruction case and in simpler no-zone segments.

RQ1 is therefore answered as a deployment trade-off. Cloud placement improved robustness in difficult visual conditions, while edge placement provided the most direct local decision path and the lowest observed timing for lightweight local inference. Model configuration shaped this balance, because larger or more capable

models did not dominate all scenarios and could introduce higher timing or runtime costs.

For RQ2, the results show clear differences in runtime burden and operational energy between edge and cloud inference. Edge inference avoided remote request queues, cloud-specific stale-result behaviour, and queue-drop counters. It also had the lower processing-energy burden in the full-day trace, with edge processing energy ranging from 0.0297 kWh to 0.0348 kWh. Cloud inference introduced transport delay, queue drops, stale results, and sensitivity to maximum in-flight request configuration, showing that cloud operation depends on request scheduling and queue policy as well as model choice.

The cloud power results show that offloading inference shifted, rather than removed, the processing-energy cost. Cloud processing combined edge-side power with logged cloud GPU power and consumed 0.2171 kWh to 0.2795 kWh over the analysed full-day window. RQ2 is therefore answered as follows: edge placement had the lower measured processing-energy burden, while cloud placement introduced communication-sensitive runtime overhead and higher measured edge-plus-cloud-GPU energy use.

For RQ3, the projected lighting results show that the zone-aware controller reduced lighting energy relative to both baseline automation strategies across all evaluated models and placements. The physical-switch baseline kept all mapped fixtures active for the full analysed period, while the motion-detector baseline remained close to continuous activation because the hold time kept lamp-bearing zones active after detected activity. The zone-aware controller reduced energy by using reconstructed zone states to activate only the red, yellow, and blue zones as needed.

After processing energy was included, all evaluated model-placement combinations still produced positive projected net savings relative to both analytical baselines. The net-energy margin was larger for edge than for cloud because edge processing consumed less energy. RQ3 is therefore answered by showing that the zone-aware controller improved projected lighting-energy and net-energy outcomes relative to the chosen baselines, with deployment placement affecting the size of the net-energy margin.

8.2 Contribution

The main contribution of this thesis is a system-level evaluation of inference placement in a zone-based, camera-supported lighting-control pipeline. Rather than evaluating person detection in isolation, the thesis shows how model and placement choices affect the full control path: from visual perception and zone assignment to operational feasibility and projected lighting outcomes.

This contribution is supported by two concrete outcomes. First, the thesis implements a shared edge/cloud comparison pipeline, making deployment placement visible as part of the control problem rather than as a separate infrastructure choice.

Second, it provides a traceable log-based evaluation workflow that connects detector behaviour and spatial assignment to reconstructed occupancy traces, projected lighting intervals, and net-energy estimates.

Together, these results provide practical guidance for AI-enabled building automation: deployment decisions should be scenario-aware, cloud inference should be treated as an operational subsystem, and processing energy and privacy exposure should be evaluated alongside perception quality.

8.3 Scope and limitations

The conclusions are limited by the scope of the study. The evaluation used one system architecture, one main sensing setup, one edge platform, one cloud execution path, a limited set of controlled traces, and one field-like normal-working-day trace. The results may change with different rooms, camera positions, lighting conditions, network paths, edge devices, model families, or fixture layouts. The full-day lighting results are analytical projections from reconstructed occupancy states and nominal fixture wattages; physical actuation, dimming behaviour, illuminance feedback, and occupant comfort were not measured.

The cloud energy comparison is also bounded. It includes logged cloud GPU power and measured edge-side power, but not complete provider-side infrastructure energy. Network robustness claims are likewise limited to the observed transport, queue, and stale-result telemetry, because no controlled degraded-network experiment was performed. These boundaries do not invalidate the findings, but they define how far the claims can be taken.

8.4 Final conclusion

Within these boundaries, the thesis shows that a stereo-depth-assisted, zone-aware perception-and-decision pipeline can produce useful projected energy benefits, but that the way inference is deployed strongly affects the quality and cost of the resulting control signal. Cloud inference improved several difficult perception cases, while edge inference provided lower processing energy, lower privacy exposure, and the fastest path for lightweight local models. The most defensible conclusion is therefore not that one placement should always be preferred, but that inference placement and model configuration must be evaluated together as architectural variables in control-oriented AI systems.

Future work should focus on the parts that remain outside the implemented runtime: physical lighting actuation, longer deployments, occupant-centred response-quality evaluation, controlled degraded-network testing, and empirically validated hybrid edge-cloud strategies. A suitable follow-up design would compare edge, cloud, and hybrid policies on the same time-synchronised traces and in a longer live deployment with fixture actuation, occupant feedback, and controlled network-impairment conditions.

Bibliography

[1] C. Ning, D. Shen, Y. Dong, Y. Wang, and P. Duan, "Fine-grained modeling and optimal control methods via video-based positioning for multi-occupant smart lighting systems," *Building and Environment*, vol. 272, p. 112683, 2025.

[2] A. Ortiz-Peña, A. Honrubia-Escribano, and E. Gómez-Lázaro, "Electricity consumption and efficiency measures in public buildings: A comprehensive review," *Energies*, vol. 18, p. 609, Jan. 2025.

[3] L. Xie, K. Shan, and S. Wang, "A generic framework and strategies for integrating AI into building automation systems for field-level optimization of HVAC systems," *Energy*, vol. 333, p. 137405, July 2025.

[4] D. G. Lema, R. Usamentiaga, and D. F. García, "Quantitative comparison and performance evaluation of deep learning-based object detection models on edge computing devices," *Integration, the VLSI Journal*, vol. 95, p. 102127, Mar. 2024.

[5] K. Kansal, Y. Wong, and M. S. Kankanhalli, "Implications of privacy regulations on video surveillance systems," *ACM Transactions on Multimedia Computing, Communications, and Applications*, vol. 21, p. 184, July 2025.

[6] U.S. Department of Energy, "Lighting controls." [Online]. Available: `https://www.energy.gov/energysaver/lighting-controls`, 2026. Accessed: 2026-02-18.

[7] S. Chun, C.-S. Lee, and J.-S. Jang, "Real-time smart lighting control using human motion tracking from depth camera," *Journal of Real-Time Image Processing*, vol. 10, pp. 805–820, 2015.

[8] S. Y. Chun and C.-S. Lee, "Applications of human motion tracking: Smart lighting control," in *2013 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 387–392, IEEE, 2013.

[9] V. Nguyen, V. Dhopate, H. Huynh, H. Bouhlal, A. Annengala, G. L. Scoccia, M. Martinez, V. Stoico, and I. Malavolta, "On-device or remote? on the energy efficiency of fetching LLM-generated content," in *2025 IEEE/ACM 4th Inter-*

national Conference on AI Engineering – Software Engineering for AI (CAIN), pp. 72–82, Apr. 2025.

[10] D. A. Miles, "Let's stop the madness part 2: Understanding the difference between limitations vs. delimitations," 2019.

[11] R. Zhang, M. Kong, B. Dong, Z. O'Neill, H. Cheng, F. Hu, and J. Zhang, "Development of a testing and evaluation protocol for occupancy sensing technologies in building hvac controls: A case study of representative people counting sensors," *Building and Environment*, vol. 208, p. 108610, 2022.

[12] SICK AG, *WSE2S-2P3230S04 W2 Photoelectric Sensors: Data Sheet*. SICK AG, Feb. 2026. Product datasheet, document generated 2026-02-13.

[13] SICK AG, *UC12-1123E UC12 Ultrasonic Distance Sensors: Data Sheet*. SICK AG, Mar. 2026. Product datasheet, document generated 2026-03-12.

[14] Infineon Technologies AG, *Ceiling-mounted occupancy detection using XENSIV™ DEMO BGT60TR13C 60 GHz radar*. Infineon Technologies AG, Feb. 2024. Application note AN141319, Revision 1.00.

[15] Signify Holding, *EasyAir SNH212 MC, SNHB212 MC Datasheet*. Signify Holding, July 2024. Datasheet for PIR-based industrial sensor.

[16] TCP, *Microwave Motion Sensor: Product Specifications*. TCP, Oct. 2022. Product specification sheet.

[17] Stereolabs, "Using the depth sensing api (zed sdk documentation)," 2026. Accessed: 2026-02-17.

[18] A. Abdelsalam, M. Mansour, J. Porras, and A. Happonen, "Depth accuracy analysis of the zed 2i stereo camera in an indoor environment," *Robotics and Autonomous Systems*, vol. 179, p. 104753, Sept. 2024.

[19] L. Li, J. Wang, S. Yang, and H. Gong, "Binocular stereo vision based illuminance measurement used for intelligent lighting with LED," *Optik: International Journal for Light and Electron Optics*, vol. 237, p. 166651, 2021.

[20] Stereolabs, "Zed 2i camera overview & datasheet." `https://cdn.sanity.io/files/s18ewfw4/production/f3860c2dfd475deb15f6d5ed28d2b106d6639d94.pdf/ZED%202i%20Datasheet%20Feb%202022.pdf`, Feb. 2022. Accessed: 2026-01-27.

[21] Q. Song, M. Kuwano, T. Obo, and N. Kubota, "Multi-property infrared sensor array for intelligent human tracking in privacy-preserving ambient assisted living," *Applied Sciences*, vol. 15, p. 12144, 2025.

[22] S. Słomiński and M. Sobaszek, "Dynamic autonomous identification and intelligent lighting of moving objects with discomfort glare limitation," *Energies*,

vol. 14, no. 21, p. 7243, 2021.

[23] T. Miller, I. Durlik, E. Kostecka, P. Kozlovska, M. Staude, and S. Sokołowska, "The role of lightweight AI models in supporting a sustainable transition to renewable energy: A systematic review," *Energies*, vol. 18, no. 5, p. 1192, 2025.

[24] S. Dikshit, R. Dixit, R. Tiwari, and P. Jain, "Performant multilingual modulated and multiplexed memory distilled model with adaptive activation ensembles," *SN Computer Science*, vol. 6, p. 644, 2025.

[25] A. A. S. Mohammad, S. I. S. Mohammad, B. Al Oraini, A. Vasudevan, A. Hindieh, A. Altarawneh, M. T. Alshurideh, and I. Ali, "Strategies for applying interpretable and explainable AI in real world IoT applications," *Discover Internet of Things*, vol. 5, p. 71, 2025.

[26] J. Nielsen, "Response times: The 3 important limits." `https://www.nngroup.com/articles/response-times-3-important-limits/`, Jan. 1993. Accessed: 2025-12-16.

[27] European Union, "Regulation (eu) 2016/679 of the european parliament and of the council of 27 april 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data (general data protection regulation)." `https://eur-lex.europa.eu/legal-content/EN/TXT/PDF/?uri=CELEX:32016R0679`, 2016. Official Journal of the European Union, OJ L 119, 4.5.2016, pp. 1–88.

[28] European Data Protection Board, "Guidelines 3/2019 on processing of personal data through video devices." `https://www.edpb.europa.eu/sites/default/files/files/file1/edpb_guidelines_201903_video_devices_en_0.pdf`, 2020. Version 2.0, adopted on 29 January 2020.

[29] European Data Protection Board, "Guidelines 05/2020 on consent under regulation 2016/679." `https://www.edpb.europa.eu/sites/default/files/files/file1/edpb_guidelines_202005_consent_en.pdf`, 2020. Version 1.1, adopted on 4 May 2020.

[30] European Data Protection Board, "Guidelines 4/2019 on article 25 data protection by design and by default." `https://www.edpb.europa.eu/sites/default/files/files/file1/edpb_guidelines_201904_dataprotection_by_design_and_by_default_v2.0_en.pdf`, 2020. Version 2.0, adopted on 20 October 2020.

[31] European Commission, "Commission implementing decision (eu) 2021/914 of 4 june 2021 on standard contractual clauses for the transfer of personal data to third countries pursuant to regulation (eu) 2016/679." `https://eur-lex.europa.eu/legal-content/EN/TXT/PDF/?uri=CELEX:32021D0914`, 2021. Official Journal of the European Union, OJ L 199, 7.6.2021, pp. 31–61.

[32] European Data Protection Board, "Recommendations 01/2020 on measures that supplement transfer tools to ensure compliance with the eu level of protection of personal data." https://www.edpb.europa.eu/system/files/2021-06/edpb_recommendations_202001vo.2.0_supplementarymeasurestransferstools_en.pdf, 2021. Version 2.0, adopted on 18 June 2021.

[33] K.-J. Stol and B. Fitzgerald, "The ABC of software engineering research," *ACM Transactions on Software Engineering and Methodology*, vol. 27, pp. 1–51, Sept. 2018.

[34] NVIDIA Corporation, *Jetson Nano Developer Kit User Guide*. NVIDIA, Jan. 2020. DA_09402_004, Version 1.3, January 15, 2020.

[35] G. Jocher and Ultralytics, "Ultralytics yolov5." https://github.com/ultralytics/yolov5, 2020. Version 7.0. Accessed: 2026-01-27.

[36] A. R. Putri, A. Irwansyah, F. Arifin, E. Purwantini, and C. K. Wijaya, "Real-time embedded vision system for road damage detection utilizing deep learning," *JOIV: International Journal on Informatics Visualization*, vol. 9, pp. 1971–1979, Sept. 2025.

[37] ALLEA – All European Academies, "The european code of conduct for research integrity – revised edition 2023," June 2023. Berlin.

A

Appendix