Edge Computing in Commercial Poultry: Minimum Artificial Intelligence Inference Requirements for Real-Time Broiler Harvest Count Verification

Engr. Aris R. Salomon, MBA, MSCS
img2data.tech
Las Piñas, Philippines
Email: ars@img2data.tech

***Abstract*—Commercial poultry harvesting in many Philippine farm settings still relies on manual bird handling, weighing, and count verification procedures. Although computer vision can support these tasks, practical use depends not only on detector capability but also on whether inference can run reliably on affordable edge hardware near the point of operation, especially where cloud connectivity is limited or unstable. This study evaluates the minimum practical hardware conditions needed to support local broiler harvest count verification using a trained YOLO-based oriented bounding box workload.**

The detector is treated as a fixed deployment workload, while the study focuses on hardware feasibility under realistic farm-side conditions. A live RTSP harvest-camera stream is processed locally while latency, achieved frames per second, memory use, processor utilization, and thermal behavior are recorded. Benchmarks on four CPU-only platforms show a wide separation in practical suitability. Effective throughput ranged from 4.44 to 24.73 frames per second in short runs and from 4.58 to 17.29 frames per second in sustained 10-minute runs. The strongest desktop platform maintained the best throughput with clear thermal margin, the laptop and mid-tier desktop remained usable but thermally strained, and the older 4-core desktop approached a practical lower bound. These results show that deployment suitability depends on sustained throughput and thermal stability rather than raw inference speed alone. The study therefore contributes a deployment-centered view of edge-computing sufficiency for broiler harvest verification.

***Index Terms*—Edge computing, computer vision, broiler harvesting, count verification, oriented bounding boxes, real-time inference.**

I. Introduction

Commercial poultry production is increasingly influenced by Precision Livestock Farming approaches that use sensing, computer vision, and machine learning to improve monitoring, documentation, and decision support across animal-production workflows [1], [2]. Recent studies show that Artificial Intelligence can already support tasks such as live broiler weight estimation, welfare-related observation, lesion assessment at slaughter, chicken counting, and dense-scene detection in commercial settings [3]–[7]. These developments suggest that poultry computer vision is maturing beyond proof-of-concept model training and toward operationally relevant farm applications.

One such application is harvest-stage count verification. Commercial broiler harvesting remains a labor-intensive and welfare-sensitive stage of production in which birds are manually caught, moved, weighed, and documented under strong time pressure [8]–[10]. In these conditions, count verification matters because it links field handling activity with inventory accountability, weighing records, and harvest reporting. A practical computer vision system for this setting is not intended to replace human workers, but to provide local verification support during a demanding manual workflow.

The technical difficulty of this use case should not be understated. Harvest-stage broiler scenes are crowded, dynamic, and visually irregular. Birds overlap heavily, head orientation varies, motion blur can occur during handling, and useful viewpoints depend strongly on camera placement and distance [11], [12]. Related poultry-monitoring studies also show that commercial environments impose difficulties for detection, tracking, and behavior interpretation because animals move unpredictably, appearance changes over time, and environmental conditions can degrade model reliability [13]–[15]. These constraints make deployment quality inseparable from both model design and operating conditions.

For that reason, detection accuracy alone is not a sufficient research endpoint. Prior work by the present author suggests that oriented bounding boxes can better represent broiler-head geometry than axis-aligned boxes in dense harvest scenes, motivating a YOLO-OBB pipeline for this study [16]. Yet even a suitable detector remains operationally limited if it cannot sustain useful inference speed, memory behavior, and thermal stability on field-deployable hardware. This is especially important in remote farm settings, where unstable connectivity can make cloud-dependent inference impractical and where low-cost local systems may be more realistic than continuously connected infrastructure [17]–[19].

The broader edge-AI literature reinforces this deployment concern. Studies in agriculture and embedded intelligence consistently report that practical adoption depends on the interaction among model size, hardware capability, latency, energy or thermal constraints, and the maturity of the inference backend [18], [20]. Even outside poultry, work on field computer vision shows that sensor choice, camera geometry, and runtime conditions can materially affect whether an otherwise strong model remains usable in real-world operation [12]. For livestock applications, recent synthesis work likewise emphasizes that robustness, deployment cost, edge efficiency, and cross-environment reliability remain central unresolved issues [1], [19].

This study therefore asks a deployment-centered question: what are the minimum Artificial Intelligence inference requirements needed to support near-real-time broiler harvest count verification on edge-computing hardware? Using a trained YOLO-based oriented bounding box detector, the work evaluates selected hardware platforms not only by whether they can run the model, but by whether they can do so at an operationally useful level of latency, throughput, utilization, thermal stability, and count-verification reliability. The emphasis is on identifying the lower practical bound for local inference rather than simply naming the fastest device.

A. Problem Context

Harvest-stage broiler scenes are visually challenging because birds overlap, orientations vary, and handling conditions are dynamic [11], [16]. These characteristics make oriented bounding box detection a suitable representation for broiler-head localization, particularly when the intended output is not just object detection but count verification at the frame and event levels. Even so, an accurate detector is not sufficient if the underlying device cannot maintain acceptable latency, throughput, or thermal stability during sustained inference, especially under practical farm-side conditions where portability, cost, and network independence matter [17], [18], [20].

B. Study Objective

The main objective of this study is to determine the minimum edge-computing conditions needed to support near-real-time local count verification during broiler harvest operations. To do this, the study benchmarks selected hardware configurations using metrics such as inference latency, frames per second, memory use, processor utilization, thermal behavior, and count-verification reliability. In this sense, the work connects poultry computer vision with the broader need for field-ready, resource-aware Artificial Intelligence deployment in agriculture and livestock systems [1], [19].

C. Contribution of the Study

This work contributes to the poultry Artificial Intelligence literature in three ways. First, it extends prior broiler computer vision research toward a deployment-centered question that is directly relevant to commercial use [5]–[7], [16]. Second, it frames edge computing as a practical enabler for local inference in connectivity-limited farm settings rather than treating deployment as an afterthought [17]–[19]. Third, it provides an evidence-based basis for choosing affordable hardware that can support farm-ready count-verification tools under realistic operational constraints.

D. Scope of the Paper

The paper focuses on local inference requirements for a harvest-stage broiler count-verification pipeline using trained YOLO-OBB models. It does not aim to replace farm workers or fully automate harvest handling. Instead, the intended role of the system is to support manual weighing and verification by providing local visual intelligence under realistic operational constraints. The scope is therefore narrower than general poultry monitoring, but more deployment-specific: the study is concerned with whether a useful broiler-harvest verification system can run locally, reliably, and affordably near the point of operation [3], [4], [10].

II. Related Work

A. Computer Vision in Poultry Monitoring

Computer vision has become an important component of Precision Livestock Farming because it enables continuous, non-invasive observation of animals in settings where manual inspection is labor-intensive, subjective, or too intermittent for real-time decision support [1], [2]. In poultry production, recent work has applied machine learning to a broad set of tasks including live-weight estimation, lesion assessment, welfare-oriented behavior monitoring, broiler discomfort analysis, health-state recognition, and automated slaughter-related inspection [3], [4], [15], [21], [22]. Taken together, these studies show that poultry vision systems are no longer limited to simple image classification problems; they increasingly aim to generate management-relevant information under commercial conditions.

This broader literature is useful for the present study in two ways. First, it confirms that poultry houses and handling environments can support meaningful video-based inference despite occlusion, clutter, lighting variation, and flock density. Second, it shows that practical value often comes from converting detections into operational indicators such as weight, welfare events, abnormality alerts, or inspection outcomes rather than treating object detection as an endpoint in itself [3], [4], [23]. That framing aligns closely with harvest-stage count verification, where the relevant output is not merely the presence of birds in an image, but whether detections can support usable verification signals during a manual workflow.

B. Broiler Detection and Harvest-Stage Monitoring

Among poultry-vision tasks, broiler detection and counting are especially relevant to this paper because they address dense scenes in which birds overlap heavily and individual separation is difficult. DFCCNet and PCCNet show that flock-level chicken counting can be learned from dense visual data, while the PIO dataset strengthens the field by providing a more realistic benchmark for broiler detection under farm conditions [5]–[7]. Related broiler studies have also extended beyond counting toward movement tracking, health recognition, thermal-comfort assessment, and video-based production monitoring, showing that commercial broiler environments can support multiple forms of useful automated analysis [13], [14], [17].

Even so, most of this literature focuses on general in-house monitoring, welfare assessment, or production analytics rather than harvest-stage verification. That distinction matters. Harvest conditions create a narrower but operationally distinct use case in which birds are manually caught, carried, weighed, and documented in dynamic short-duration events. The visual problem is therefore not identical to routine flock observation inside a house. Prior work by the present author helps bridge this gap by showing that oriented bounding boxes can better represent broiler-head geometry than standard axis-aligned boxes in dense scenes, and by identifying practical camera-placement considerations for high-density broiler monitoring [11], [16]. These studies provide a direct methodological foundation for the present paper's YOLO-OBB harvest-verification pipeline.

C. Edge Computing for Agricultural Artificial Intelligence

Edge computing is increasingly discussed as a practical enabler for agricultural Artificial Intelligence because it allows inference to be performed near the point of sensing instead of depending entirely on cloud transfer and remote processing [18], [19]. This shift is especially relevant in livestock operations and remote farm settings where connectivity may be weak, intermittent, or operationally inconvenient. In such environments, the value of an AI model depends not only on predictive accuracy, but also on whether it can run locally with acceptable latency, memory use, thermal behavior, and power demands.

Existing edge-oriented studies support this perspective but also reveal a gap. Poultry-specific work has shown that lightweight detection pipelines can be adapted for local chicken-health inference [17]. Outside poultry, agricultural edge-AI studies demonstrate that low-cost devices can support practical computer-vision tasks when lightweight models and appropriate hardware pairings are used, but they also show large performance differences among edge platforms such as Raspberry Pi class devices, Coral-assisted systems, and Jetson-based accelerators [18], [20]. More generally, livestock-focused synthesis work emphasizes that robust deployment requires attention to efficiency, model compression, hardware awareness, environmental variability, and the trade-off between accuracy and operational cost [1], [19]. These observations are directly relevant to the present study, which treats deployment feasibility as a primary research question rather than a final implementation detail.

D. Operational and Welfare Context

The harvest stage must also be understood in its operational and welfare context. Manual catching, carrying, loading, and transport are known to influence stress exposure, handling quality, and labor efficiency in broiler production [8]–[10]. This means that harvest is not simply another imaging scenario; it is a time-sensitive production event in which interventions must remain practical, non-disruptive, and compatible with human-led work. A local count-verification system has potential value precisely because it can add visibility and documentation without requiring major workflow redesign.

This context also helps explain why the present problem cannot be reduced to generic poultry detection. A harvest-support tool must be deployable close to the point of operation, function reliably under visually difficult handling conditions, and provide outputs that are meaningful to farm-side documentation. In that sense, the literature on catching welfare and manual harvest procedures defines the operational constraints within which any proposed edge-AI solution must prove useful [8]–[10].

E. Research Gap

Existing literature therefore establishes four important points: poultry computer vision is already useful for several production and welfare tasks; broiler scenes can be analyzed under dense commercial conditions; edge-AI can make local agricultural inference practical; and harvest operations impose their own welfare and workflow constraints [1], [5], [6], [8], [17]. However, a clear gap remains at the intersection of these themes. There is still limited evidence on the minimum hardware conditions required to run a harvest-stage broiler verification pipeline in near real time on field-appropriate devices.

In particular, the literature does not yet clearly answer which lower-cost or moderately capable edge platforms can sustain a YOLO-based harvest-verification workflow with acceptable latency, throughput, and operational reliability. Most prior studies emphasize either model design, general monitoring capability, or task-specific accuracy rather than the lower bound of practical inference requirements for a narrow farm-side use case. This study addresses that gap by evaluating deployment feasibility using hardware-centered benchmark criteria alongside count-verification reliability, with the goal of identifying the minimum practical edge-computing conditions for real-time broiler harvest verification.

III. Methodology

A. Overall Approach

This study adopts a deployment-centered methodology for evaluating whether a harvest-stage broiler count-verification system can operate in near real time on field-appropriate edge hardware. The methodological focus is not limited to detector accuracy in isolation. Instead, the work evaluates the full local inference pathway: visual input acquisition, YOLO-OBB-based broiler-head detection, post-processing, count-verification output generation, and device-level performance measurement during execution.

The basic premise is that a farm-side verification system is only useful if it satisfies two conditions simultaneously. First, it must produce detections that remain usable for count verification in dense, dynamic harvest scenes. Second, it must do so on hardware that can sustain the required computation under realistic runtime constraints. The methodology therefore combines computer-vision evaluation with systems benchmarking so that deployment feasibility is assessed as an interaction between model behavior and hardware behavior rather than as a purely algorithmic question [6], [7], [18], [19].

B. Detection Pipeline

The core detection pipeline consists of five stages: input acquisition, preprocessing, YOLO-OBB inference, detection filtering, and count-verification output generation. Input data are derived from harvest-stage images or replayed video frames captured near the weighing and handling workflow, where birds appear under partial overlap, irregular orientation, and frequent short-duration movement. These characteristics motivate the use of oriented bounding boxes, which better preserve localized head geometry in dense scenes than standard axis-aligned boxes [16], [24].

During preprocessing, input frames are resized to the target inference resolution required by the trained model while preserving a consistent evaluation protocol across devices. The model is then executed using a fixed weights file and fixed inference settings for each benchmark run. To ensure comparability, parameters such as image size, confidence threshold, non-maximum suppression behavior, and batch size are held constant unless a separate controlled comparison is explicitly performed. This design keeps hardware comparisons focused on runtime capability rather than configuration drift.

After raw detections are produced, post-processing is applied to suppress low-confidence or redundant predictions and to retain only the detections relevant to broiler-head counting. The filtered detections are then translated into count-verification outputs. At the frame level, the system produces a count estimate for each processed image or frame. At the event level, frame outputs are aggregated over a short handling interval so that the pipeline supports verification of a harvest action rather than only isolated still-image counting. This distinction is important because farm-side verification depends on temporally usable outputs, not only per-frame object localization.

C. Benchmarking Perspective

This work treats deployment feasibility as a systems question. Accordingly, the benchmark process is designed to characterize both instantaneous and sustained device behavior under repeated inference. For each selected edge platform, the pipeline is executed locally using identical test inputs and a standardized runtime procedure. Measurements are collected for inference latency, achieved frames per second, memory consumption, processor or accelerator utilization, and thermal behavior. These variables were chosen because they jointly determine whether a device is merely capable of running the detector or can sustain useful farm-side operation over time [17], [18], [20].

Latency is used to quantify per-inference responsiveness, while achieved frames per second captures the effective throughput available for near-real-time use. Memory use is monitored because low-cost edge devices can fail practically through memory pressure even when nominal compute capability appears sufficient. Processor or accelerator utilization is recorded to indicate how fully the hardware is being driven during inference and to help interpret bottlenecks. Thermal behavior is included because sustained performance degradation, throttling, or instability can make an apparently fast device unsuitable for use during actual harvest operations.

These hardware-centered measurements are paired with count-verification reliability so that devices are not judged only by speed. A platform that attains high throughput but produces unstable or inconsistently usable count outputs under replayed harvest events is not considered operationally sufficient. The benchmark perspective therefore links technical performance with application usefulness.

D. Evaluation Outputs

Two categories of outputs are central to the study:

1) hardware performance metrics, such as latency, throughput, utilization, and thermal stability
2) verification performance metrics, such as frame-level and event-level count consistency

Hardware performance metrics are summarized using repeated-run measurements and sustained-operation observations. Depending on the final experiment configuration, reported statistics may include mean latency, median latency, dispersion measures, effective frames per second, peak and average memory use, processor load, accelerator load, and temperature traces over test duration. These outputs describe the runtime envelope within which the detector operates on each device.

Verification performance metrics are designed to reflect the actual role of the system in harvest support. At minimum, the study evaluates frame-level count agreement and event-level count consistency across replayed harvest sequences. Where annotated reference counts are available, this evaluation can be expressed through count error statistics or agreement measures between predicted and expected count behavior. The purpose is not to replace a full detector-validation study, but to confirm that the runtime behavior of the system remains operationally meaningful while being executed on constrained hardware.

This combined evaluation allows the study to identify the minimum practical inference conditions for field deployment. In this context, "minimum" does not mean the cheapest device that can produce any output; it means the lowest practical hardware condition that can maintain sufficiently responsive, stable, and verification-relevant local inference during the intended workflow.

E. Reproducibility Considerations

Several controls are used to improve reproducibility. The same trained model weights, input set, preprocessing routine, and post-processing parameters are used across tested devices unless a controlled ablation explicitly requires otherwise. Each device is benchmarked with a documented software stack that includes operating system, Python environment, deep learning framework version, inference backend, and any hardware-specific acceleration libraries. Device specifications such as processor class, memory capacity, storage type, cooling condition, and accelerator availability are also recorded so that later interpretation of performance differences remains grounded in actual hardware characteristics.

To reduce experimental drift, benchmark runs should follow a fixed sequence that includes model loading, warm-up inference, repeated timed execution, and sustained-operation measurement over a predefined interval. Logging procedures, sampling frequency, and metric definitions should remain consistent across devices as far as the platform allows. This is particularly important in edge-AI comparison studies, where differences in backend maturity or thermal management can otherwise obscure the distinction between model limitations and device limitations [18]–[20].

Finally, the methodology intentionally separates model-development questions from deployment-feasibility questions. The trained detector is treated as a fixed workload that represents the intended harvest-verification task. By holding the workload stable and varying the execution platform, the study can more clearly determine which edge-computing conditions are sufficient, marginal, or inadequate for real-time broiler harvest count verification.

IV. Edge Hardware Setup

A. Benchmark Scope

The study uses a small but practically relevant set of locally available x86 devices spanning a stronger laptop-class reference platform, a strong desktop platform, a mid-tier desktop platform, and an older lower-bound desktop platform. The purpose is not to compare idealized edge appliances, but to identify practical hardware conditions achievable with systems that are realistically obtainable in the local setting.

At the current stage, four systems have completed the full benchmark workflow: a Lenovo Legion 5 laptop used as the reference platform, a Dell OptiPlex Micro 7020 with an Intel Core i7-14700T, a Dell OptiPlex 3080 with an Intel Core i5-10500T, and a Dell OptiPlex 3040 with an Intel Core i5-6500T. Together, these systems already define a meaningful range from strong desktop execution down to a weak lower-bound case, so any remaining devices are optional extensions rather than essential benchmarks for the present paper.

B. Deployment Assumptions

The evaluated systems are treated as on-site inference units executing the trained YOLO-OBB workload locally, with no dependence on remote cloud inference during operation. This assumption is central because poultry-farm environments cannot always rely on stable high-bandwidth connectivity for continuous video processing.

The deployment model also assumes a human-led workflow rather than a fully automated industrial line. Accordingly, the device does not need to behave like a server-class platform, but it must process harvest-stage frames quickly enough to provide timely count-verification support during or immediately after manual weighing events.

C. Runtime Environment and Logging

The benchmark environment used Python 3.12.9 on the Legion 5, Python 3.14.5 on the OptiPlex 7020 and OptiPlex 3040, and Python 3.14.3 on the OptiPlex 3080, with OpenCV, Ultralytics, and the same portable benchmark workflow on all four systems. The runner recorded stream resolution, reported source frame rate, read failures, per-frame read time, per-frame inference time, full cycle time, and periodic system telemetry. Additional machine details such as integrated graphics and storage configuration were preserved in the device profiles, but they are not emphasized here because the confirmed experiments were executed in CPU-only mode.

The telemetry stream included overall CPU usage, process-level CPU usage, resident memory, system memory state, and temperature readings. This logging strategy is important because a device that appears fast in raw timing but exhibits unstable stream behavior, excessive temperature, or sustained processor saturation may still be a poor deployment choice.

V. Experiments

A. Benchmark Design

The present experimental stage focuses on confirmed benchmarking of four edge-computing platforms under a fixed local inference workload. The detector is treated as a constant deployment task rather than as the main object of comparison. This design keeps the experiment aligned with the paper's central question: whether practical edge hardware can sustain useful local broiler-harvest inference under realistic operating conditions.

The visual input for the confirmed benchmark was a live RTSP stream from the harvest-camera setup, captured at a reported resolution of 1920×1080 and a reported stream rate of 30 frames per second. The benchmark runner was configured to process the stream as fast as possible, with no artificial sampling cap, so that the results would reflect the actual local throughput of each device under the chosen workload.

B. Runtime Configuration

The workload used the primary deployment-oriented YOLO26s-OBB benchmark weight with a fixed image size of 640 pixels and fixed post-processing settings. The confirmed runs were executed in CPU-only mode on all four machines. This choice was intentional because the immediate objective at this stage was to establish a comparable CPU baseline for local deployment feasibility across devices with different form factors, cooling conditions, and processor generations.

The benchmark runner also collected periodic system telemetry alongside per-frame inference metrics. This telemetry included system CPU usage, process CPU usage, resident memory, system memory state, CPU temperature, and lightweight GPU telemetry when available. Although the current workload was CPU-bound, retaining the telemetry channel was useful for confirming whether the platforms remained stable under prolonged execution.

C. Live Overlay Example

Figure 1 shows a representative live-inference frame from a dense harvest scene. It is presented in operational form rather than as a cleaned visualization because the study is concerned with practical deployment under real farm conditions. The frame confirms that live detection is running, but it also shows that heavy label overlap can reduce operator-facing readability in crowded scenes.

TABLE I
CONFIRMED BENCHMARK PLATFORMS USED IN THE STUDY.

Platform	CPU	RAM	OS / kernel	Role in study
Lenovo Legion 5	11th Gen Intel Core i7-11800H	16.5 GB	Fedora 43 / 7.0.10	Laptop reference platform; CPU-only benchmark execution
Dell OptiPlex Micro 7020	Intel Core i7-14700T	33.3 GB	Fedora 43 / 7.0.12	Strong desktop platform
Dell OptiPlex 3080	Intel Core i5-10500T	16.5 GB	Fedora 44 / 6.19.2	Mid-tier desktop platform
Dell OptiPlex 3040	Intel Core i5-6500T	16.7 GB	Fedora 44 / 7.0.11	Older lower-bound desktop platform

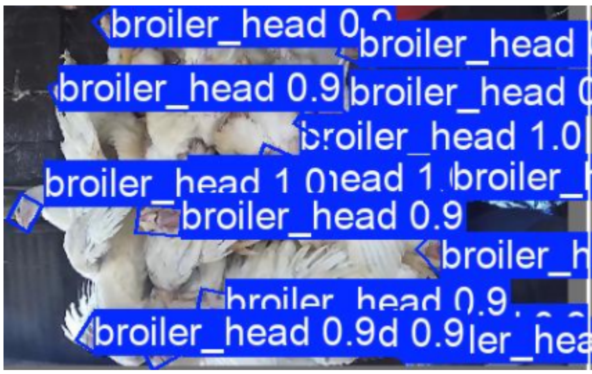

Fig. 1. Live RTSP inference output in a dense broiler-harvest scene. The figure shows the actual operator-facing overlay produced by the edge-deployed YOLO-OBB pipeline. Detections are generated successfully, but overlapping labels and boxes reduce visual clarity and highlight a practical usability constraint.

TABLE II
SHORT-RUN BENCHMARK COMPARISON

Metric	Legion	7020	3080	3040
Measured frames	300	300	300	300
Mean infer. time (s)	0.05875	0.03849	0.09156	0.22115
Mean cycle time (s)	0.05950	0.03906	0.09220	0.22198
Effective FPS	16.24	24.73	10.57	4.44
Mean RSS memory	1.03 GB	1.05 GB	1.03 GB	0.96 GB
Mean CPU temp.	95.67 C	51.99 C	96.0 C	57.93 C

TABLE III
SUSTAINED 10-MINUTE BENCHMARK COMPARISON

Metric	Legion	7020	3080	3040
Measured frames	8115	10375	6132	2749
Mean infer. time (s)	0.07299	0.05688	0.09706	0.21707
Mean cycle time (s)	0.07384	0.05776	0.09771	0.21786
Effective FPS	13.52	17.29	10.22	4.58
Mean RSS memory	1.04 GB	1.07 GB	1.03 GB	0.97 GB
Mean CPU temp.	96.47 C	58.03 C	96.0 C	63.31 C

D. Initial Short-Run Benchmark

The first confirmed benchmark pass on each machine was designed as a short full-speed validation run. It used 3 warm-up frames followed by 300 measured frames. This run was intended to verify that the RTSP pipeline, fixed local inference workload, and telemetry logging all operated correctly under live input conditions before the longer sustained test was attempted.

All four short runs completed successfully with 303 total frames observed and 300 measured frames recorded after warm-up. No read failures occurred on any machine. The strongest short-run result was produced by the OptiPlex 7020 at 24.73 effective frames per second, followed by the Legion 5 at 16.24, the OptiPlex 3080 at 10.57, and the OptiPlex 3040 at 4.44.

E. Sustained 10-Minute Benchmark

After the short validation runs, the same fixed workload was executed as a sustained 10-minute benchmark on all four machines. These second runs retained the same live RTSP source, CPU-only execution mode, image size, and general inference configuration, but replaced the frame-limited stop condition with an explicit 600 s wall-clock runtime limit. The purpose of the sustained run was to evaluate not only throughput, but also stability, thermal behavior, and resource persistence over a more realistic deployment interval.

In the sustained 10-minute runs, the same performance order remained. The OptiPlex 7020 achieved 17.29 effective frames per second, the Legion 5 achieved 13.52, the OptiPlex 3080 achieved 10.22, and the OptiPlex 3040 achieved 4.58. No read failures occurred during any sustained run.

F. Experimental Value of the Current Stage

Although the full device inventory has not been exhausted, the confirmed four-platform experiments already establish a reliable benchmark workflow and reveal the main runtime behaviors needed for deployment-centered evaluation. In particular, they show that the pipeline can ingest live RTSP data stably, sustain repeated local inference, and emit the latency, throughput, memory, and thermal measurements required by the study.

VI. RESULTS AND DISCUSSION

A. Comparative Throughput

The benchmark results show a clear performance separation across the four tested CPU-only platforms. In both short and sustained runs, the same ranking remained: OptiPlex 7020 first, Legion 5 second, OptiPlex 3080 third, and OptiPlex 3040 last. Effective throughput ranged from 24.73 to 4.44 frames per second in short runs and from 17.29 to 4.58 frames per second in sustained runs. This ranking establishes a practical hardware gradient from clearly strong desktop execution down to a weak lower-bound case.

The four-platform comparison also confirms that brief benchmarks can overstate field performance, although the effect differs by machine. The Legion 5 dropped by roughly 16.7% from its short-run effective rate to its sustained rate, the OptiPlex 7020 dropped by roughly 30.1%, and the OptiPlex 3080 dropped by only about 3.3%. The OptiPlex 3040 also changed only slightly between the short and sustained runs because it was already operating near its practical ceiling even in the short benchmark. This is the more useful interpretation for deployment: sustained throughput matters more than best-case startup speed, and once hardware falls into the 10 FPS range the workload may still be useful for some delayed or light verification settings, whereas 4 to 5 FPS begins to look clearly marginal for practical near-real-time use.

B. Stream Stability and Bottleneck Location

The benchmark results also make clear that video input acquisition is not the main performance bottleneck on any confirmed platform. Mean read time stayed below 1 ms in all short and sustained runs, and no read failures occurred in any condition. On the Legion 5, mean read time ranged from 0.00073 s to 0.00083 s. On the OptiPlex 7020, it ranged from 0.00056 s to 0.00087 s. On the OptiPlex 3080, it ranged from 0.00063 s to 0.00065 s. On the OptiPlex 3040, it ranged from 0.00077 s to 0.00081 s. These values are negligible relative to end-to-end inference cycle time.

This distinction matters for the overall research question. If stream ingestion had been unstable, the paper would need to treat input transport as a major deployment issue. Instead, the current evidence suggests that the key edge-computing challenge on these tested systems is sustained compute performance, not RTSP acquisition overhead. That finding becomes even stronger after adding the OptiPlex 3080 and the older OptiPlex 3040, because both machines show stable RTSP handling despite materially lower throughput than the strongest desktop.

The live overlay example in Fig. 1 adds a complementary observation. Even when inference is running correctly and stream ingestion remains stable, dense harvest scenes can produce heavy label overlap that reduces operator-facing readability. Deployment sufficiency therefore depends not only on execution speed, but also on whether the resulting visualization remains usable during real work.

C. Processor Load and Thermal Behavior

The most important comparative finding is the interaction between throughput and thermal behavior. The Legion 5 and OptiPlex 3080 both operated at roughly 96°C on average, with peaks reaching 99–100°C. By contrast, the OptiPlex 7020 remained much cooler at about 52–58°C, while the OptiPlex 3040 remained near 58–63°C. The thermal evidence shows that the middle-tier OptiPlex 3080 behaves more like the thermally stressed laptop than like the cooler OptiPlex 7020, despite delivering substantially lower throughput than the Legion 5.

Across all four systems, the workloads remained distinctly CPU-intensive. On the Legion 5, mean overall CPU utilization increased from 50.83% in the short run to 55.32% in the sustained run, while process-level CPU utilization increased from 758.56% to 815.58%. Corresponding increases were also observed on the OptiPlex 7020, where mean overall CPU utilization rose from 26.59% to 29.69% and process-level CPU utilization from 733.91% to 816.69%, and on the OptiPlex 3080, where mean overall CPU utilization increased from 65.37% to 68.51% and process-level CPU utilization from 776.56% to 813.95%. The OptiPlex 3040 exhibited the highest overall CPU demand, with mean overall CPU utilization remaining near saturation at 90.00% in the short run and 91.89% in the sustained run, while process-level CPU utilization increased from 265.39% to 276.69%. It should be noted that process-level CPU utilization values above 100% reflect aggregated usage across multiple CPU cores and therefore indicate parallel CPU consumption rather than measurement error. Collectively, these results confirm substantial processor involvement across all platforms and support the conclusion that the workload is fundamentally compute-bound.

For the edge-computing framing of the paper, this is a central finding. A device may satisfy near-real-time throughput requirements and still remain only marginally suitable for repeated field deployment if it must operate near thermal limits to do so. Conversely, a device may remain thermally comfortable but still be insufficient if its throughput is too low. The Legion 5 appears computationally sufficient but thermally strained, the OptiPlex 7020 appears both faster and substantially more thermally comfortable, the OptiPlex 3080 appears computationally moderate but also thermally strained, and the OptiPlex 3040 appears thermally manageable but computationally weak for this workload.

D. Memory Behavior

Memory behavior was substantially less problematic than throughput or temperature on all confirmed platforms. Resident memory stayed near 1 GB throughout all runs. Mean RSS on the Legion 5 was approximately 1.03 GB in the short test and 1.04 GB in the sustained run, while the OptiPlex 7020 used approximately 1.05 GB and 1.07 GB respectively, the OptiPlex 3080 used approximately 1.03 GB in both modes, and the OptiPlex 3040 used approximately 0.96 GB and 0.97 GB. Peak RSS remained close to 1.06 GB on the laptop, 1.09 GB on the newer desktop, 1.05 GB on the OptiPlex 3080, and 0.99 GB on the older desktop. These values indicate that the workload has a fairly stable memory footprint across different CPUs and runtime durations.

This suggests that memory is not the limiting factor on any tested device under the present workload. That observation is important because it helps separate different forms of deployment risk. Across the four machines, the limiting factors are compute strength and thermal sustainability rather than memory capacity. This is a more informative conclusion than simply reporting that the pipeline "ran successfully."

E. Interpretation for Edge-Computing Sufficiency

Taken together, the confirmed results support a more precise deployment judgment than the earlier single-platform view allowed. All four currently tested machines are capable of sustaining local harvest-stage inference without cloud dependence in the narrow sense of continued execution and stable RTSP handling. However, they do not represent equally good deployment conditions. The laptop reference platform shows meaningful slowdown between the short and sustained runs while operating near its thermal ceiling, the newer desktop platform sustains higher throughput with much larger thermal headroom, the OptiPlex 3080 occupies a moderate middle tier at about 10 FPS sustained but also operates at very high temperature, and the older OptiPlex 3040 remains far below the other three in processing speed.

Accordingly, the Legion 5 should currently be interpreted as a feasible but thermally stressed reference case, the OptiPlex 7020 is the stronger provisional deployment candidate among the devices tested so far, the OptiPlex 3080 represents a moderate middle case whose practicality is weakened by high sustained temperature, and the OptiPlex 3040 functions as an early lower-bound warning case. This comparison makes clear that practical edge deployment cannot be evaluated by latency alone. Sustained heat, cooling margin, and long-run throughput stability are decisive variables, but so is the simple question of whether a device can stay above a practically useful processing rate.

F. Implications for the Remaining Study

These results sharpen the next steps of the research. If additional devices are tested later, they should use the same workload and logging procedure so that any refinement of the boundary between “clearly usable” and “marginal” hardware remains directly comparable. However, the present four-platform comparison already shows the shape of that transition: the workload is strong on a newer high-core desktop, thermally constrained on a powerful laptop, moderate but thermally stressed on a 6-core mid-tier desktop, and only weakly practical on an older 4-core desktop.

The current results do not close the study, but they establish its direction. They show that the benchmark workflow is valid, that the workload is genuinely CPU-bound, that RTSP ingest is stable, and that thermal headroom and sustained throughput are decisive variables in identifying the minimum practical edge-computing condition for the full harvest-verification pipeline.

VII. Conclusion

This study treats broiler harvest verification as an edge-computing deployment problem rather than primarily as a detector-development problem. Using a trained YOLO-OBB workload executed on a live RTSP harvest-camera stream, the confirmed results show a clear spread across four CPU-only platforms. The strongest desktop platform achieved 24.73 and 17.29 effective frames per second in the short and sustained runs, the laptop reference platform achieved 16.24 and 13.52, the mid-tier desktop achieved 10.57 and 10.22, and the older 4-core desktop achieved only 4.44 and 4.58.

The results also show that throughput alone is not enough to judge deployment suitability. The laptop reference platform and the mid-tier desktop both operated above 95°C on average, whereas the strongest desktop remained much cooler and the older 4-core desktop remained cooler but substantially slower. Memory behavior remained manageable and stream ingestion remained stable on all four machines. The main differentiating constraints are therefore sustained compute capability and thermal headroom rather than capture reliability or memory exhaustion.

The current results therefore establish a credible comparative baseline without requiring every remaining available machine to be benchmarked. They show that the workload is strong on a newer high-core desktop, thermally stressed on the tested laptop, moderate but still thermally strained on a 6-core desktop, and already close to a practical lower bound on an older 4-core desktop. Among the tested systems, the OptiPlex 7020 is the stronger provisional edge-computing candidate because it delivers the best sustained throughput with clear thermal margin. Future work may extend the same benchmark protocol to additional optional devices and determine more precisely where local broiler-harvest inference shifts from operationally sufficient to marginal or impractical under sustained farm-side conditions.

Acknowledgment

The author thanks Dra. Luisa E. Dandan for her guidance and technical perspective in poultry science throughout the development of this study. Her insights on broiler production, harvest procedures, poultry-house practice, and animal-welfare considerations helped keep the work anchored in actual field conditions.

The author also acknowledges Ms. Genalin C. Salomon for her assistance in dataset preparation and annotation review. Her contribution to labeling and checking broiler-head instances supported the construction of the harvest-stage oriented bounding box dataset used in this research.

Appreciation is likewise extended to Mr. Amiel Jose Miguel C. Salomon for his support in field preparation, data gathering, and deployment-related technical setup. His assistance contributed to obtaining usable visual data and maintaining practical study conditions during collection activities.

The author further thanks Mr. Angelo Raya, the Raya family, and the farm personnel who permitted and supported on-site data collection in their commercial poultry operation. Their cooperation and accommodation during field visits and overnight collection work made it possible to carry out the study in a real harvest-stage environment.

References

[1] D. Distante, C. Albanello, H. Zaffar, S. Faralli, and D. Amalfitano, “Artificial intelligence applied to precision livestock farming: A tertiary study,” *Smart Agricultural Technology*, vol. 11, p. 100889, 2025.

[2] S. Neethirajan, “Rethinking poultry welfare—integrating behavioral science and digital innovations for enhanced animal well-being,” *Poultry*, vol. 4, 2025.

[3] M. Campbell, P. Miller, K. Díaz-Chito, S. Irvine, M. Baxter, J. Martínez del Rincón, X. Hong, N. McLaughlin, T. Arumugam, and N. O’Connell, “Automated precision weighing: Leveraging 2d video feature analysis and machine learning for live body weight estimation of broiler chickens,” *Smart Agricultural Technology*, vol. 10, p. 100793, 2025.

[4] N. van Staaveren, K. van Langeveld, J. Schulte-Landwehr, J. Mulder, P. Galliot, P. Créach, P. Sztandarski, J. Marchewka, B. Forkman, N. Van Noten, F. Tuyttens, M. Giersberg, M. Manet, and T. Rodenburg, "Validation of an automated system for at-slaughter assessment of footpad dermatitis and hock burn in broiler chickens," *Poultry Science*, vol. 105, p. 106968, 2026.

[5] J. Lv, J. Wang, C. Peng, and Q. Huang, "Dfccnet: A dense flock of chickens counting network based on density map regression," *Animals*, vol. 13, no. 23, p. 3729, 2023.

[6] Y. Guo, Z. Wu, Z. Su, J. Zhao, and X. Li, "Pccnet: A point supervised dense chickens flock counting network," *Smart Agricultural Technology*, vol. 10, p. 100795, 2025.

[7] K. Boniche, E. Cruz, J. C. Rangel, M. Hidalgo-Rodríguez, and F. Gómez-Donoso, "Pio, a large-scale dataset for broiler chicken detection under real poultry farming conditions," *Scientific Data*, 2026.

[8] K. E. Kittelsen, E. G. Granquist, A. L. Aunsmo, R. O. Moe, and E. Tolo, "An evaluation of two different broiler catching methods," *Animals*, vol. 8, no. 9, p. 141, 2018.

[9] I. Wolff, S. Klein, E. Rauch, M. Erhard, J. Mönch, and S. Härtle, "Harvesting-induced stress in broilers: Comparison of a manual and a mechanical harvesting method under field conditions," *Applied Animal Behaviour Science*, vol. 221, p. 104877, 2019.

[10] M. A. Hettmannsperger and I. Ruhnke, "Manual catching and transportation of poultry with a focus on chickens and european practices," *Poultry*, vol. 5, 2026.

[11] A. R. Salomon, "Machine learning in commercial poultry farming: Optimizing camera placement to bridge ai performance and monitoring in high-density broiler stocking," *TechRxiv*, Apr. 2025.

[12] P. J. Hennessy, T. J. Esau, A. W. Schumann, Q. U. Zaman, K. W. Corscadden, and A. A. Farooque, "Evaluation of cameras and image distance for cnn-based weed detection in wild blueberry," *Smart Agricultural Technology*, vol. 2, p. 100030, 2022.

[13] H. Brunet, P. Creach, and D. Concordet, "Optimal estimation of broiler movement for commercial tracking," *Smart Agricultural Technology*, vol. 3, p. 100113, 2023.

[14] A. J. de Oliveira Júnior, G. d. S. Sousa, E. Dal Pai, O. C. P. de Almeida, M. Mollo Neto, R. P. Simões, and S. R. L. de Souza, "System for assessing broilers thermal comfort," *Smart Agricultural Technology*, vol. 1, p. 100007, 2021.

[15] L. Yang and J. Song, "Real-time detection of dead broilers in high-density poultry houses using high-resolution video and semantic segmentation," *Poultry Science*, vol. 105, p. 106969, 2026.

[16] A. R. Salomon, "Machine learning in commercial poultry farming: Comparing standard and oriented bounding boxes in yolov11 for broiler detection," *TechRxiv*, Apr. 2025.

[17] Q. Tong, J. Wang, and W. Yang, "Edge ai-enabled chicken health detection based on enhanced fcos-lite and knowledge distillation," *arXiv preprint arXiv:2407.09562*, 2024.

[18] Y. Majeed, M. O. Ojo, and A. Zahid, "Standalone edge ai-based solution for tomato diseases detection," *Smart Agricultural Technology*, vol. 9, p. 100547, 2024.

[19] S. Neethirajan, "Green ai for livestock perception: Efficient architectures, multimodal sensing, and robust deployment," *TechRxiv*, 2025.

[20] M. N. Jacob, "Lightweight machine learning for edge devices: Classical baselines vs. tinycnn with int8 quantization," *TechRxiv*, 2025.

[21] T. Bergamasco, A. Ambrosi, V. Tregnaghi, R. Urbani, G. Nalesso, F. Menegon, A. Trocino, M. Pravato, F. Bordignon, S. Sparesato, G. Manca, and G. Di Martino, "Precision livestock farming: Yolov12-based automated detection of keel bone lesions in laying hens," *Poultry*, vol. 4, 2025.

[22] N. Coimbra da Silva, I. d. A. Nääs, J. d. S. G. Barros, and D. J. d. Moura, "Modeling broiler discomfort under commercial housing: Seasonal trends and predictive insights for precision livestock farming," *Poultry*, vol. 4, 2025.

[23] D. B. Jensen, M. Toscano, E. van der Heide, M. Grønvig, and F. Hakansson, "Comparison of strategies for automatic video-based detection of piling behaviour in laying hens," *Smart Agricultural Technology*, vol. 10, p. 100745, 2025.

[24] K. Wang, Z. Wang, Z. Li, A. Su, X. Teng, M. Liu, and Q. Yu, "Oriented object detection in optical remote sensing images using deep learning: A survey," *arXiv preprint arXiv:2302.10473*, 2024.