

Article

Comparative Analysis of Real-Time Detection Models for Intelligent Monitoring of Cattle Condition and Behavior

Oleg Ivashchuk¹, Zhanat Kenzhebayeva^{1,*} , Alexey Zhigalov², Moldir Allaniyazova¹ , Gulnara Kazyieva¹, Kaiyrbek Makulov¹ , Vyacheslav Fedorov³ and Olga Ivashchuk³

¹ Department of Computer Science, Faculty of Science and Technology, Caspian University of Technology and Engineering (KUTI) Named After Sh. Yessenov, Micro District 32, Aktau 130000, Kazakhstan; oleg.ivashuk@gmail.com (O.I.); moldir.allaniyazova@yu.edu.kz (M.A.); gulnara.kazyieva@yu.edu.kz (G.K.); kaiyrbek.makulov@yu.edu.kz (K.M.)

² Individual Entrepreneur Zhigalov A.A. (neural.dev), Agmashenebeli Str., 1, Batumi 6004, Adjara, Georgia; jgaloff@gmail.com

³ Department of Information and Robotic Systems, Belgorod State National Research University, 85, Pobedy St., Belgorod 308015, Russia; fedorov_v@bsuedu.ru (V.F.); olga.ivashuk@mail.ru (O.I.)

* Correspondence: zhanat.kenzhebayeva@yu.edu.kz; Tel.: +7-(701)-125-61-23

Abstract

This study benchmarks nine state-of-the-art object detection models on a specialized cattle dataset to assess accuracy and inference speed for real-time agricultural applications. Using a unified protocol without model-specific augmentations, and evaluating all detectors on identical RTX 4090 hardware, we provide a fair architectural comparison of two-stage, one-stage, and transformer-based models. D_FINE_L and Co_DETR_R_50 achieved the highest accuracy ($AP@[0.50:0.95] = 0.872$ and 0.851), while RTMDet and YOLOv11_L were the fastest (15.81 and 19.14 ms/image). All models showed substantial accuracy gains on the domain dataset compared to COCO, while maintaining consistent relative speed rankings.

Keywords: benchmarking; object detection; agriculture; real time; RTMDet_m; D_FINE_L; D_FINE_L; MMDetection



Academic Editors: Chih-Lung Lin, Bor-Jiunn Hwang, Shaou-Gang Miaou and Chi-Hung Chuang

Received: 21 October 2025

Revised: 21 November 2025

Accepted: 26 November 2025

Published: 2 December 2025

Citation: Ivashchuk, O.; Kenzhebayeva, Z.; Zhigalov, A.; Allaniyazova, M.; Kazyieva, G.; Makulov, K.; Fedorov, V.; Ivashchuk, O. Comparative Analysis of Real-Time Detection Models for Intelligent Monitoring of Cattle Condition and Behavior. *Algorithms* **2025**, *18*, 763. <https://doi.org/10.3390/a18120763>

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

1.1. Relevance of the Problem

Modern agriculture is undergoing intensive integration of digital technologies into all aspects of business and production processes. In this context, computer vision and deep learning models play a crucial role in enabling highly accurate, non-invasive object detection—an essential component in livestock management. Automated intelligent animal monitoring systems make it possible not only to accurately count livestock but also to track their health, analyze behavior and growth rates, optimize feeding schedules, and detect diseases in time.

For example, computer vision systems can detect early signs of illness or injury—such as nasal discharge or eye problems—allowing farmers to take timely action and prevent disease outbreaks [1]. Monitoring animal movement and interactions helps identify signs of stress or discomfort [2]. Moreover, drones equipped with cameras provide effective aerial surveillance of livestock across large fields, helping track grazing patterns and locate missing animals [3].

For all these applications, detection capability is of critical importance, as it directly affects operational efficiency, economic viability, and animal welfare. However, this domain

presents several unique challenges: significant variation in animal sizes (from small objects in drone footage to close-up large animals), frequent occlusions within herds, complex dynamic natural backgrounds, and a strict need for real-time processing for practical farm management decisions.

Despite the widespread use of deep learning models, there is still a noticeable lack of systematic comparative benchmarks focused on real-time commercial performance for domain-specific agricultural datasets. This gap is especially critical under the constraints of real-time operation and commercial deployment, which require balancing accuracy, inference speed, resource efficiency, and licensing considerations.

Thus, conducting a systematic benchmarking of modern open-source object detectors on a specialized animal dataset, with a focus on both accuracy and inference speed under real-time and commercial constraints, is a highly relevant and timely research task.

1.2. Research Problem Statement

To address this need, we design a systematic benchmarking study that targets domain-specific agricultural conditions rather than general-purpose benchmarks. While COCO (Common Objects in Context, Microsoft) remains the de facto standard for evaluating object detectors [4], it is often insufficient for specialized applications due to differences in object characteristics (e.g., number of classes, unique shapes), scene complexity, and performance requirements (e.g., strict timing constraints, accuracy thresholds).

As mentioned above, in agricultural settings, objects may be very small or very large, frequently occluded, and appear against backgrounds that differ substantially from urban or domestic scenes represented in COCO. Therefore, directly applying models trained on general datasets without domain adaptation and specialized benchmarking can lead to inadequate performance in real-world agricultural conditions.

The goal of this study is to conduct systematic benchmarking of nine modern open-source object detection architectures on a specialized cattle dataset, evaluating performance in terms of accuracy and inference speed under real-time and commercial conditions.

We hypothesize that the accuracy of detectors fine-tuned on our dataset will generally correlate with their known performance on public datasets (e.g., COCO), though deviations may occur due to the domain-specific nature of the cattle dataset. In contrast, inference speed is expected to remain consistent with previously reported values.

This study will analyze the performance of various architectural paradigms—two-stage, one-stage, and transformer-based—on a real, domain-specific dataset. Particular attention is given to the trade-off between accuracy and inference speed, which is critical for practical real-time applications. Domain adaptation aspects affecting the generalization capability of models in agricultural environments are also considered.

The expected outcomes—a unique set of empirical benchmarks and practical recommendations—will provide tangible value for stakeholders in the agro-industrial sector, helping them make informed decisions when selecting suitable object detection models for specific needs. Detailed documentation of the experimental setup and methodology underscores the importance of research reproducibility.

2. Overview of Existing Object Detectors

2.1. General Information on Object Detectors

Modern deep-learning-based object detection systems are typically grouped into three major architectural paradigms: two-stage, one-stage, and transformer-based detectors. The rapid progress of convolutional neural networks (CNNs) has substantially advanced 2D object detection, positioning CNNs as the foundation of most modern vision-based recognition systems [5].

Object detectors are widely used across many domains, including agriculture, robotics, and urban monitoring, and their architectures are broadly categorized into two-stage and one-stage pipelines [6].

2.1.1. Two-Stage Detectors

Two-stage detectors, such as the R-CNN family, operate by first generating region proposals and then refining and classifying them in a second stage [6]. This multi-step pipeline offers high localization accuracy but typically results in lower inference speed.

Cascade R-CNN extends the two-stage paradigm through a sequence of detectors trained at progressively increasing IoU thresholds, improving localization precision and robustness to false positives [7].

The architecture has been widely adopted and is available in open-source implementations, including those provided by the MMDetection framework [8].

2.1.2. One-Stage Detectors

One-stage detectors predict object locations and categories in a single pass, offering reduced latency and improved suitability for real-time applications [9].

RetinaNet

RetinaNet addresses the critical problem of foreground–background imbalance through the introduction of Focal Loss, which down-weights easy negatives and focuses learning on hard examples [10].

To support multi-scale detection—a key requirement for identifying objects of diverse sizes—it incorporates Feature Pyramid Networks (FPN), enabling high-quality recognition of both small and large targets [11].

RTMDet

RTMDet is designed as a high-efficiency real-time detector, incorporating optimized convolutional blocks that reduce computational complexity while retaining strong accuracy [12].

Its architecture is scalable and supports extensions to segmentation and rotated object detection tasks, and its reference implementations are publicly available [13].

YOLOv11

One of the newest additions to the YOLO family, YOLOv11, introduces architectural improvements aimed at balancing speed and accuracy in high-density or multi-scale environments [14].

Key innovations include lightweight feature extraction modules, enhanced multi-scale fusion blocks, and refined spatial attention mechanisms.

An overview of the YOLOv11 architecture and its performance characteristics is presented in the recent literature [15].

2.1.3. Transformer-Based Detectors

Transformer-based object detectors represent a new paradigm that leverages attention mechanisms to capture global image context, achieving state-of-the-art results across many benchmarks [16].

Recent comprehensive reviews highlight the evolution of transformer-based detection architectures and their applications in vision systems [17].

DETR

DETR (Detection Transformer) introduced a fully end-to-end detection pipeline that removes the need for anchor boxes and non-maximum suppression, establishing the foundation for transformer-based object detection [16,17].

D-FINE

D-FINE enhances DETR-type models by redefining bounding-box regression as a fine-grained distribution refinement task and by incorporating self-distillation techniques for improved localization accuracy [18].

DINO

DINO further improves DETR by introducing denoising-based training, enhanced query initialization, and multi-scale feature integration, achieving robust convergence and high detection accuracy [19].

RT-DETR

RT-DETR, presented as a real-time detection transformer, demonstrates that DETR-style models can surpass traditional CNN-based detectors in real-time applications by leveraging hybrid encoders and multi-scale attention [20].

Co-DETR

Finally, Co-DETR incorporates collaborative hybrid assignment strategies and auxiliary detection heads, significantly improving the performance of transformer-based detectors on small, overlapping, or occluded objects [21].

3. Research Methodology

Building on the architectural overview in Section 2, this section details the experimental pipeline used to benchmark the selected detectors, including dataset construction, hardware–software configuration, model adaptation, training protocol, and evaluation metrics, with an emphasis on reproducibility.

3.1. Data Preparation

For this study, several public datasets containing cattle images were collected and aggregated: “Cattle (Cows) Object Detection Dataset,” “Animals Detection Images Dataset,” “Cattle Breeds Dataset,” and “Bovine Cattle Images.” All redundant and irrelevant data (non-cow images) were removed, along with duplicates and mirrored duplicates. Manual label correction was performed to ensure high annotation accuracy.

As a result, 4367 images containing 14,400 bounding box annotations were obtained.

To ensure dataset representativeness, the data were stratified by scene type (e.g., “small distant cows” and “dense herds”), allowing proportional representation of each group in training and validation sets.

All annotations were converted into COCO JSON format, compatible with MMDetection, with strict validation of data integrity and file path consistency.

As a result, 4367 images containing 14,400 bounding box annotations were obtained. To ensure dataset representativeness, the data were stratified by scene type (e.g., “small distant cows” and “dense herds”), allowing proportional representation of each group in training and validation sets. All annotations were converted into COCO JSON format, compatible with MMDetection, with strict validation of data integrity and file path consistency. The aggregated and cleaned dataset (4367 images, 14,400 annotations) is publicly available for academic research and deployment purposes

at <https://huggingface.co/datasets/AlexeyZhigalov/Detectors> (accessed on 21 November 2025). All code configurations, fine-tuning scripts, and inference results are hosted in an open-source repository at https://github.com/nrldev/Free_detectors (accessed on 21 November 2025), ensuring full reproducibility of the experimental methodology.

3.2. Hardware and Software Configuration

All experiments were conducted on the following documented hardware:

GPU: NVIDIA RTX 4090 (24 GB VRAM);

CPU: AMD Ryzen 9 9900X;

RAM: 64 GB.

The software environment was configured in an isolated Conda environment running Linux with the following:

NVIDIA driver version 570.86.10;

CUDA Toolkit 12.8;

PyTorch 2.4.1;

MMDetection 3.3.0.

All dependencies were recorded in a requirements.txt file to ensure reproducibility.

3.3. Model Selection and Adaptation

Nine state-of-the-art object detection models were selected for comparative analysis:

Cascade R-CNN [8], RetinaNet [22], RTMDet [23], D-FINE [13], DETR [24], DINO [25], RT-DETR [26], Co-DETR [27], and YOLOv11 [28].

The main selection criterion was model size (20–70 M parameters)—large enough for robust accuracy but lightweight enough for real-time and embedded applications.

MMDetection-based models (Cascade R-CNN, RetinaNet, RTMDet, DETR, DINO, RT-DETR, Co-DETR) were obtained from the official repository under the Apache-2.0 license, allowing free academic and commercial use.

YOLOv11 (from Ultralytics) is distributed under AGPL-3.0, requiring derivative works to remain open-source.

D-FINE and its checkpoints were obtained from the official repository (<https://github.com/Peterande/D-FINE>, accessed on 21 November 2025), also licensed under Apache-2.0.

All configuration files were modified to match the single-class detection task (“cow”), with the following:

num_classes = 1;

input resolution fixed at 1280×1280 ;

unified logging and checkpoint storage (work_dirs).

The unified augmentation strategy is as follows:

To ensure fairness, all model-specific augmentations were replaced with a shared minimal set:

Horizontal flip ($p = 0.5$);

Random rotation (-10° to $+10^\circ$);

Filtering out objects smaller than 16×16 px.

This minimized variability between models while maintaining comparability, although it may have slightly reduced the potential accuracy of certain models (since aggressive augmentations often improve generalization).

3.4. Training Protocol

To ensure a rigorous and reproducible comparison, we followed a unified training protocol for all architectures. Each model was initialized from the baseline configuration recommended by its original authors within the MMDetection or Ultralytics framework and

then empirically fine-tuned to the specific properties of the “Cattle” dataset, particularly the single-class setup and the fixed input resolution of 1280×1280 pixels.

Optimization Strategy:

We employed specific optimizers tailored to each model architecture (e.g., AdamW for transformer-based models like D-FINE and Co-DETR, and SGD for classic two-stage detectors like Cascade R-CNN). The choice of optimizers and initial learning rates (LR) was validated through preliminary runs to ensure stable convergence.

The optimization process was critical for ensuring that all models adhered to the unified augmentation strategy (Section 3.3). This rigor involved specific troubleshooting for models like YOLOv11, where initial configurations did not correctly apply the minimal shared augmentations, leading to sub-optimal initial results. The final optimized configuration in Table 1 reflects the settings necessary to ensure maximum performance under the established fair benchmarking protocol, resulting in the improved metrics observed in the results section.

Table 1. Optimized hyperparameters and fine-tuning configuration of detection models.

Model	Optimizer	Base LR	Scheduler	Weight Decay	Best Epochs/ Epochs	Batch Size
Cascade_R_CNN_R_50_FPN	SGD	0.005	MultiStepLR	0.0001	13/20	4
Co_DETR_R_50	AdamW	2.5×10^{-5}	CosineAnnealing	0.05	17/20	2
DETR_R_50	AdamW	1×10^{-5}	MultiStepLR	0.0001	72/75	4
DINO_R_50	AdamW	0.001	MultiStepLR	0.05	24/25	2
RetinaNet_R_50_FPN	SGD	0.001	MultiStepLR	0.0001	10/24	4
RT_DETR_R_50vd	AdamW	1×10^{-6}	CosineAnnealing	0.05	32/36	4
RTMDet_m	AdamW	1×10^{-6}	CosineAnnealing	0.05	17/36	8
YOLOv11_L	Auto (SGD/AdamW)	-	-	-	-/120	8
D_FINE_L	AdamW	0.00025	CosineAnnealingLR	0.000125	13/36	4

Learning Rate Scheduling:

Training stability was managed using standard LR decay strategies. For most CNN-based models, a “Step” or “Cosine Annealing” scheduler was used, with a linear warm-up phase during the first 500–1000 iterations to prevent gradient explosion.

Convergence Criteria:

All models were trained for a fixed number of epochs sufficient to reach convergence, defined as the point where the validation loss stabilized and the Average Precision (AP) metric ceased to improve over 10 consecutive epochs. The specific hyperparameters used for the final training runs of each model are detailed in Table 1.

Model Selection:

To mitigate the effects of stochasticity, the final evaluation was performed using the checkpoint that achieved the highest AP@[0.5:0.95] on the validation set, rather than simply taking the last epoch’s weights.

3.5. Model Performance Evaluation Protocol

Performance evaluation included two key aspects: accuracy and inference speed.

(1) Accuracy (AP) [29]:

Each model's best checkpoint was evaluated using the COCOeval script on a 569-image test set (input size 1280×1280 , no Test-Time Augmentation).

Metrics recorded the following:

AP@[IoU = 0.50:0.95] (main metric);

AP@[IoU = 0.50];

AP for small, medium, and large objects;

Average Recall (AR).

(2) Inference Speed (Latency):

A custom benchmarking script measured latency (ms/image) on an NVIDIA RTX 4090 (FP32 precision, batch size = 1).

Warm-up runs were performed to stabilize performance before timing.

Both preprocessing and postprocessing times (resize + normalization) were included to reflect real-world conditions.

Two measurement modes were implemented:

Reproducible mode:

cudnn.benchmark = False, deterministic = True.

Prioritize consistency for academic comparison.

Production-optimized mode:

cudnn.benchmark = True; deterministic = False.

Reflects realistic deployment performance for industrial systems.

Both latency values are reported in the results—the first ensuring reproducibility, the second representing operational speed.

4. Results

Using the benchmarking protocol described in Section 3, we evaluated nine object detection models on the specialized cattle dataset. The results cover both accuracy metrics (Average Precision, AP) and inference speed (latency and FPS), and they are also compared with the models' reported performance on the COCO dataset.

The results include both accuracy metrics (Average Precision, AP) and inference speed metrics (latency and FPS) for each model, along with comparisons to their reported performance on the COCO dataset.

4.1. Evaluation of Detection Accuracy and Inference Speed

The final accuracy and inference speed results for each model are summarized in Table 2.

Table 2. Accuracy and Inference Speed of Cattle Detection Models.

Model	AP@[0.5:0.95]	AP@0.50	AR@[0.5:0.95]	Speed (ms/img, R)	FPS (R)	Speed (ms/img, P)	FPS (P)	Params (M)
Cascade_R_CNN_R_50_FPN	0.772	0.926	0.798	35.83	27.91	39.81	25.12	69.15
Co_DETR_R_50	0.851	0.966	0.906	280.94	3.56	283.67	3.53	64.45
DETR_R_50	0.777	0.947	0.847	24.70	40.49	25.17	39.72	41.55
D_FINE_L	0.872	0.966	0.909	21.61	46.28	22.23	44.98	30.66
DINO_R_50	0.819	0.949	0.898	77.99	12.82	79.11	12.64	47.54
RetinaNet_R_50_FPN	0.751	0.941	0.807	26.11	38.30	28.38	35.24	36.33
RT_DETR_R_50vd	0.722	0.843	0.838	30.12	33.20	30.67	32.61	42.78
RTMDet_m	0.773	0.944	0.842	15.81	63.24	15.73	63.56	24.66
YOLOv11_L	0.758	0.900	0.798	19.14	52.24	19.74	50.65	25.31

Note: The AP values are derived from the COCOeval evaluation results. The inference speed (ms/img) and frames per second (FPS) are reported for both "reproducible" (R) and "production-optimized" (P) modes.

To enable a fair comparison, the total number of parameters for each model was also calculated, providing an additional measure of model complexity.

Accuracy was evaluated using Average Precision (AP) at multiple IoU thresholds, while inference speed was measured as the average time per image (ms/image) and frames per second (FPS) in two operation modes:

Reproducible (R): optimized for experimental reproducibility,

Production (P): optimized for real-world deployment performance.

4.2. Comparative Analysis

The analysis reveals that the D-FINE model achieved the highest accuracy among all tested detectors, reaching $AP@[0.5:0.95] = 0.872$ and $AP@0.50 = 0.966$.

This demonstrates its superior localization precision and high reliability in predictions.

Co-DETR also achieved strong results ($AP@[0.5:0.95] = 0.851$, $AP@0.50 = 0.966$), ranking second overall in terms of detection accuracy.

This essential trade-off between speed and precision, which is critical for real-time commercial applications in the agro-industrial sector, is visually summarized in Figure 1a,b.

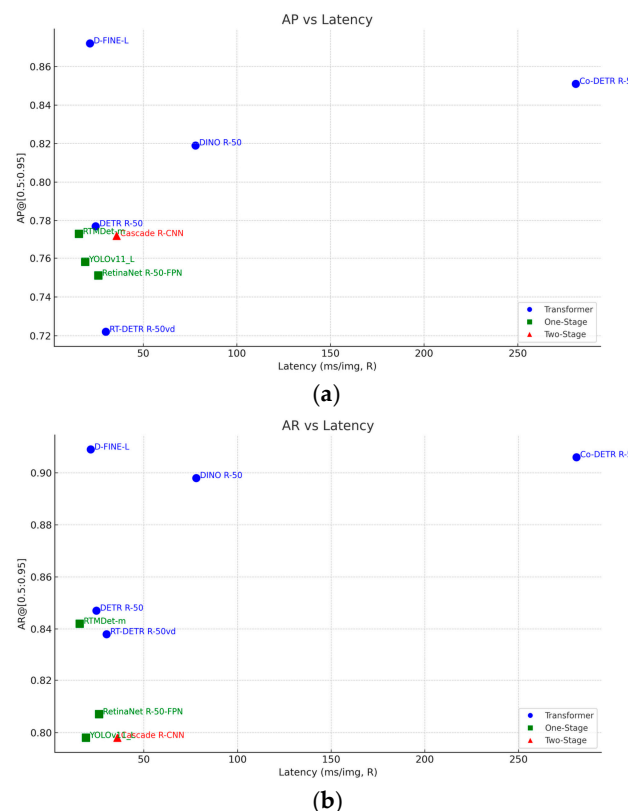


Figure 1. Accuracy–Latency Trade-off for Cattle Detection Models. (a) Average Accuracy ($AP@[0.5:0.95]$) vs. Inference Rate (ms/image). (b) Average Recall ($AR@[0.5:0.95]$) vs. Inference Rate (ms/image).

Other transformer-based models—DETR and DINO—showed good, though slightly lower, performance ($AP@[0.5:0.95] = 0.777$ and 0.819 , respectively), likely due to their architectural characteristics and domain adaptation specifics.

In terms of inference speed under reproducible conditions, RTMDet demonstrated the lowest latency—15.81 ms per image (63.24 FPS)—making it the fastest among all models.

YOLOv11 maintained its position as the second fastest model, recording 19.14 ms (52.24 FPS). Following configuration optimization (Table 1), its accuracy significantly improved to $AP@[0.5:0.95] = 0.758$. This enhanced performance strengthens its competitiveness, positioning YOLOv11 above the accuracy achieved by RetinaNet (0.751) and

RT-DETR (0.722). Both RTMDet and YOLOv11 belong to the one-stage family, confirming their optimization for speed, which is essential for real-time agro-industrial monitoring.

Despite its superior accuracy, D-FINE maintained impressive performance at 21.61 ms (46.28 FPS), making it particularly attractive for real-world applications requiring both precision and responsiveness.

Considering its balance of accuracy and speed, D-FINE emerges as the most well-rounded solution, combining top-tier precision with near real-time inference speed.

For applications where maximum speed is critical, RTMDet or YOLOv11 are preferable, as they provide acceptable accuracy with minimal latency.

Meanwhile, DETR and RetinaNet offer moderate inference times (~25–26 ms per image), suitable for use cases with less stringent real-time constraints.

However, Co-DETR (~280.9 ms) and DINO (~78.0 ms) were significantly slower, largely due to their more complex architectures and higher parameter counts, which reduce suitability for time-sensitive deployments.

4.3. Comparison with COCO Dataset

The proposed hypothesis suggested that the accuracy of object detectors on the specialized cattle dataset would correlate with their known performance on widely used benchmarks (e.g., COCO), with possible deviations due to domain specificity, while the inference speed should remain more stable.

When comparing the obtained AP@[0.5:0.95] values on the cattle dataset with the published AP metrics on COCO (for the same model versions and checkpoints), the following pattern is observed (here, COCO AP refers to the AP metric on the COCO dataset, and Cow AP refers to the AP metric on the cattle dataset—the domain-specific dataset):

Cascade R-CNN: COCO AP = 41 compared to Cow AP = 0.772—significant increase in AP on the domain dataset;

RetinaNet: COCO AP = 39.5 compared to Cow AP = 0.751—likewise, a substantial improvement;

RTMDet: COCO AP = 49.4 compared to Cow AP = 0.773—notable increase;

D-FINE: COCO AP = 55.1 compared to Cow AP = 0.872—the highest absolute improvement;

DETR: COCO AP = 39.9 compared to Cow AP = 0.777—significant improvement;

DINO: COCO AP = 50.1 compared to Cow AP = 0.819—considerable improvement;

RT-DETR: COCO AP = 53.1 compared to Cow AP = 0.722—improvement, though less pronounced compared to other transformer-based models;

Co-DETR: COCO AP = 54.8 compared to Cow AP = 0.851—strong improvement;

YOLOv11: COCO AP = 53.4 compared to Cow AP = 0.758—improvement, now showing better domain adaptation capability.

Overall, a strong positive correlation is observed: models that demonstrated high accuracy on COCO generally achieved high results on the cattle dataset.

However, the absolute AP values on the cattle dataset are significantly higher than on COCO.

This confirms the first part of the hypothesis regarding correlation, while also highlighting noticeable deviations in absolute values.

These deviations can be explained by domain-specific factors:

The cattle dataset contains only one object class, simplifying the classification task compared to the multi-class COCO dataset.

Additionally, cattle images tend to have less diverse backgrounds or more uniform poses, making detection easier than in COCO's complex and varied contexts.

Regarding inference speed, measurements in both the “reproducible” and “production” modes showed consistent similarity, confirming the reliability of the benchmarking protocol.

Comparison with the published COCO inference speeds is difficult due to hardware differences used in the original benchmarks (e.g., D-FINE and RT-DETR were tested on T4 GPUs).

Nevertheless, the relative ranking of model speeds generally remains consistent:

One-stage detectors (RTMDet, YOLOv11) remain the fastest;

Some transformer-based models (DINO, Co-DETR) are the slowest.

4.4. Qualitative Analysis

A qualitative evaluation was performed through visual inspection of detection results across several test images.

This analysis revealed not only quantitative differences but also qualitative aspects such as successful detections, false positives, and missed objects.

Visualizations showed that high-accuracy models like D-FINE and Co-DETR generally exhibit the following:

More precise bounding box localization;

Fewer false positives;

Better performance in dense herds and complex background scenes.

Figure 2 provides a visual example of the precise bounding box localization and reliable performance of high-accuracy models (D-FINE) in dense herd scenarios.

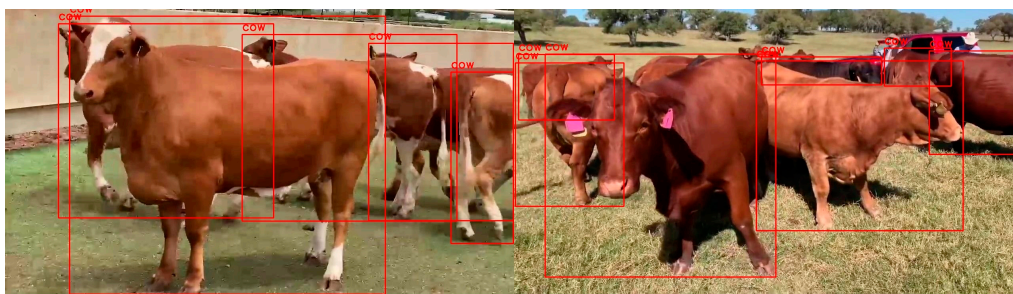


Figure 2. An example of successful cattle detection in dense scenes. The qualitative evaluation demonstrates the high localization accuracy of the model (e.g., D-FINE) when dealing with complex conditions, including partial occlusions.

For example, D-FINE successfully detected even small, distant cattle, a key advantage for aerial imagery or drone-based monitoring.

Conversely, faster models such as YOLOv11 and RT-DETR, while highly efficient, occasionally missed small or partially occluded animals, and sometimes produced false positives on background elements resembling cattle (e.g., rocks, shadows, or machinery parts).

In dense herd scenes, some detectors struggled to distinguish individual animals, leading to overlapping bounding boxes or missed detections.

False detections typically arose from background textures similar to cows, while missed detections occurred under strong lighting, motion blur, or occlusion conditions.

This qualitative analysis underscores that while quantitative metrics (e.g., AP) are crucial, visual assessment provides valuable insight into model robustness under real-world conditions and helps identify the types of domain-specific errors each model tends to make.

5. Discussion

The empirical results reported in Section 4 reveal substantial performance differences among the nine open-source object detection models evaluated on the specialized

cattle dataset, highlighting the critical importance of domain adaptation for real-world commercial applications in the agricultural sector.

5.1. Accuracy and Domain Adaptation

The results clearly demonstrate that all models achieved significantly higher Average Precision (AP) on the cattle dataset compared to their commonly reported performance on COCO. This confirms the first part of our hypothesis—the presence of a strong correlation with overall performance, accompanied by notable deviations in absolute values.

This increase in accuracy can be largely attributed to several domain-specific characteristics:

The dataset contains only a single object class (“cow”), simplifying the classification task relative to the multi-class structure of COCO.

The homogeneity of objects (cattle) and the less diverse visual context allow more effective fine-tuning.

This key observation holds strategic significance for the agri-industrial sector: even models showing moderate performance on general-purpose datasets can reach very high detection reliability and accuracy after appropriate domain adaptation (e.g., D-FINE $AP@[0.5:0.95] = 0.872$).

5.2. Synthesis of the Accuracy–Speed Trade-Off

An analysis of the trade-off between accuracy (AP) and inference speed (ms/image), as visualized in Figure 1, enables conclusions regarding the suitability of different architectural paradigms for commercial deployment.

Accuracy Leaders (Transformer Models):

D-FINE (AP = 0.872) and Co-DETR (AP = 0.851) achieved the highest accuracy values. This indicates that transformer-based architectures, through their attention mechanisms, better capture the global context of the scene and effectively deal with localization challenges in dense herds and occlusion-heavy environments—an observation further supported by qualitative analysis.

Speed Leaders (One-Stage CNNs):

RTMDet (15.81 ms/image) and YOLOv11 (19.14 ms/image) remain the fastest models. Their single-pass inference pipeline makes them ideal for high-throughput systems where latency is critical (e.g., drone-based video analysis or high-speed conveyor monitoring).

The Issue of Non-Optimized Transformers:

Despite their strong accuracy, models such as DINO (77.99 ms) and especially Co-DETR (280.94 ms) were too slow for strict real-time scenarios. This underscores the fact that the high computational cost of global attention mechanisms requires significant optimization before commercial deployment.

5.3. Architectural Interpretation for Deployment

The analysis shows that architecture selection must be strategic and aligned with the required balance between accuracy and speed.

Transformer Optimization (D-FINE and RT-DETR):

The success of D-FINE (AP = 0.872 at 21.61 ms) demonstrates that transformer architectures can be optimized to maintain top-tier accuracy while achieving near-real-time speed. This makes them highly suitable for precision-critical applications (e.g., health monitoring or behavioral analysis), where accurate bounding box localization is essential.

CNN Efficiency (RTMDet and YOLOv11):

RTMDet and YOLOv11 confirmed their speed advantage. They are the most suitable candidates for high-speed monitoring systems where a minor loss in accuracy (compared to transformer models) is compensated by maximum throughput.

Two-Stage Models:

Cascade R-CNN demonstrated solid accuracy, but its multi-step nature results in higher computational cost, making it less competitive in speed-constrained scenarios.

5.4. Stability of Inference Speed

The second part of our hypothesis—stability of inference speed—was confirmed. The relative ranking of model speeds remained consistent across all experiments and across both measurement regimes (“Reproducible” and “Production-optimized”). The minimal differences observed between regimes indicate that model speed is determined primarily by architecture and hardware configuration rather than dataset domain characteristics, in contrast to accuracy metrics.

6. Conclusions

In summary, this study conducted a systematic benchmarking of nine open-source object detection models on a specialized cattle dataset, providing critical empirical evidence for their potential commercial deployment in the agro-industrial sector.

It has been demonstrated that domain adaptation significantly increases the accuracy of all tested models compared to their reported performance on widely used benchmarks such as COCO.

6.1. Key Findings

This study introduced a robust, domain-specific benchmarking protocol, which ensured a fair, purely architectural comparison of nine SOTA models by neutralizing the effects of model-specific data augmentation strategies, thereby providing unique empirical insights into real-time agricultural applications.

This study adheres to high standards of reproducibility: the aggregated dataset (4367 images, 14,400 annotations) and all custom scripts used for fine-tuning and evaluation are available in public repositories for researchers and industry stakeholders.

D-FINE achieved the highest overall accuracy ($AP@[0.5:0.95] = 0.872$) and demonstrated strong inference speed (21.61 ms/image), making it the optimal choice for most commercial applications requiring high reliability and real-time detection.

RTMDet and YOLOv11 confirmed their benefits to be speed (15.81 ms/image and 19.14 ms/image, respectively), making them preferable for scenarios where minimal image processing latency is critical, though they achieved slightly lower accuracy compared to D-FINE.

Transformer-based architectures, such as D-FINE and Co-DETR, showed potential for achieving very high accuracy, while other models like DINO and Co-DETR remained speed-constrained despite acceptable precision.

Inference speed across all models proved to be stable and predictable, confirming that performance is primarily determined by model architecture and hardware, rather than by dataset domain.

6.2. Directions for Future Work

Investigate data augmentation strategies specifically tailored for agricultural imagery and assess their impact on model performance.

Expand the dataset to include a broader range of scenarios, lighting conditions, and possibly other livestock species.

Evaluate models on diverse hardware platforms, including cost-effective GPUs and edge accelerators, to determine their suitability for different commercial deployment environments.

Integrate object detection with tracking systems to enable behavioral analytics and automated herd management.

Explore model quantization and distillation techniques to further enhance inference speed without substantial loss of accuracy.

6.3. Recommendations for the Agro-Industrial Sector

Prioritize D-FINE for precision-critical applications.

For tasks requiring both high accuracy and real-time performance—such as detailed health monitoring and behavior analysis—D-FINE offers the best balance of precision and speed.

Use RTMDet or YOLOv11 for high-speed monitoring systems.

In scenarios where processing speed is paramount (e.g., large-scale herd counting using drones, where a slight accuracy loss is acceptable), RTMDet or YOLOv11 are the most suitable choices.

Adopt rigorous benchmarking protocols.

When selecting models for real-time deployment, organizations should rely on benchmarks that use unified augmentation strategies to ensure that performance differences are genuinely attributable to the model architecture (as demonstrated in this study), rather than pipeline-specific preprocessing advantages.

Always perform domain adaptation.

Fine-tuning models on domain-specific datasets is essential, as it leads to a substantial increase in detection accuracy compared to directly applying models pre-trained on generic datasets.

Conduct pilot deployments before scaling.

Before large-scale rollout, it is critical to run pilot studies that consider specific hardware and network environments, ensuring that the desired performance is achieved under real-world operational conditions.

Author Contributions: Conceptualization, O.I. (Oleg Ivashchuk), Z.K. and V.F.; methodology, O.I. (Oleg Ivashchuk), A.Z. and V.F.; software, A.Z., O.I. (Oleg Ivashchuk) and K.M.; validation, O.I. (Oleg Ivashchuk), A.Z., Z.K. and V.F.; formal analysis, O.I. (Oleg Ivashchuk) and A.Z.; investigation, O.I. (Oleg Ivashchuk), M.A., G.K. and K.M.; resources, Z.K., G.K., V.F. and O.I. (Olga Ivashchuk); data curation, A.Z., M.A. and K.M.; writing—original draft preparation, O.I. (Oleg Ivashchuk); writing—review and editing, Z.K., A.Z., V.F. and O.I. (Olga Ivashchuk); visualization, O.I. (Oleg Ivashchuk) and A.Z.; supervision, Z.K. and V.F.; project administration, Z.K.; funding acquisition, Z.K. and O.I. (Oleg Ivashchuk). All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Science Committee of the Ministry of Science and Higher Education of the Republic of Kazakhstan, grant number AP23489215. The APC was funded by the Science Committee of the Ministry of Science and Higher Education of the Republic of Kazakhstan under the same grant (AP23489215).

Data Availability Statement: The aggregated and cleaned cattle dataset generated and analyzed in this study is publicly available on Hugging Face at: <https://huggingface.co/datasets/AlexeyZhigalov/Detectors> (accessed on 21 November 2025). All model configurations, training scripts, evaluation tools and inference results used in this research are openly available in the GitHub repository: https://github.com/nrldev/Free_detectors (accessed on 21 November 2025). No additional restrictions apply.

Conflicts of Interest: Author Alexey Zhigalov is an Individual Entrepreneur. The remaining authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

References

1. Reza, M.N.; Lee, K.-H.; Habineza, E.; Samsuzzaman, K.; Choi, Y.K.; Kim, G.; Chung, S.-O. RGB-based machine vision for enhanced pig disease symptoms monitoring and health management: A review. *J. Anim. Sci. Technol.* **2025**, *67*, 17–42. [CrossRef] [PubMed]
2. Tsai, Y.-C.; Hsu, J.-T.; Ding, S.-T.; Rustia, D.J.A.; Lin, T.-T. Assessment of dairy cow heat stress by monitoring drinking behaviour using an embedded imaging system. *Biosyst. Eng.* **2020**, *199*, 97–108. [CrossRef]
3. Barbedo, J.G.A.; Koenigkan, L.V.; Santos, T.T. Detection of cattle using drones and convolutional neural networks. *Sensors* **2018**, *18*, 2048. [CrossRef] [PubMed]
4. Lin, T.-Y.; Maire, M.; Belongie, S.; Bourdev, L.; Girshick, R.; Hays, J.; Perona, P.; Ramanan, D.; Zitnick, C.L.; Dollár, P. Microsoft COCO: Common Objects in Context. In *Computer Vision—ECCV 2014*; Springer: Berlin/Heidelberg, Germany, 2014; pp. 740–755. [CrossRef]
5. Guarnido-Lopez, P.; Pi, Y.; Tao, J.; Mendes, E.D.M.; Tedeschi, L.O. Computer vision algorithms to help decision-making in cattle production. *Anim. Front.* **2025**, *14*, 11–22. [CrossRef] [PubMed]
6. Ruan, J.; Cui, H.; Huang, Y.; Li, T.; Wu, C.; Zhang, K. A review of occluded objects detection in real complex scenarios for autonomous driving. *Green Energy Intell. Transp.* **2023**, *2*, 100092. [CrossRef]
7. Cai, Z.; Vasconcelos, N. Cascade R-CNN: High quality object detection and instance segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.* **2021**, *43*, 1483–1498. [CrossRef] [PubMed]
8. OpenMMLab. (n.d.). Cascade R-CNN [Software Repository]. GitHub. Available online: https://github.com/open-mmlab/mmdetection/tree/main/configs/cascade_rcnn (accessed on 21 November 2025).
9. Wang, C.-Y.; Bochkovskiy, A.; Liao, H.-Y.M. YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Vancouver, BC, Canada, 17–24 June 2023. [CrossRef]
10. Lin, T.-Y.; Goyal, P.; Girshick, R.; He, K.; Dollár, P. Focal loss for dense object detection. *IEEE Trans. Pattern Anal. Mach. Intell.* **2020**, *42*, 318–327. [CrossRef] [PubMed]
11. Lin, T.-Y.; Dollár, P.; Girshick, R.; He, K.; Hariharan, B.; Belongie, S. Feature pyramid networks for object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 936–944. [CrossRef]
12. Lyu, C.; Zhang, W.; Huang, H.; Zhou, Y.; Wang, Y.; Liu, Y.; Zhang, S.; Chen, K. RTMDet: An empirical study of designing real-time object detectors. *arXiv* **2022**. [CrossRef]
13. OpenMMLab. (n.d.). RTMDet [Software Repository]. GitHub. Available online: <https://github.com/open-mmlab/mmdetection/tree/main/configs/rtdetr> (accessed on 21 November 2025).
14. He, L.; Zhou, Y.; Liu, L.; Cao, W.; Ma, J.H. Research on object detection and recognition in remote sensing images based on YOLOv11. *Sci. Rep.* **2025**, *15*, 14032. [CrossRef] [PubMed]
15. Khanam, R.; Hussain, M. YOLOv11: An overview of the key architectural enhancements. *arXiv* **2024**. [CrossRef]
16. Shehzadi, T.; Hashmi, K.A.; Stricker, D.; Afzal, M.Z. Object detection with transformers: A review. *arXiv* **2023**. [CrossRef] [PubMed]
17. Yu, L.; Tang, L.; Mu, L. A review of detection transformer: From basic architecture to advanced developments and visual perception applications. *Sensors* **2025**, *25*, 3952. [CrossRef] [PubMed]
18. Peng, Y.; Li, H.; Wu, P.; Zhang, Y.; Sun, X.; Wu, F. D-FINE: Redefine regression task in DETRs as fine-grained distribution refinement. *arXiv* **2024**. [CrossRef]
19. Zhang, H.; Li, F.; Liu, S.; Zhang, L.; Su, H.; Zhu, J.; Ni, L.M.; Shum, H.-Y. DINO: DETR with improved denoising anchor boxes for end-to-end object detection. *arXiv* **2022**. [CrossRef]
20. Zhao, Y.; Lv, W.; Xu, S.; Wei, J.; Wang, G.; Dang, Q.; Liu, Y.; Chen, J. DETRs beat YOLOs on real-time object detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 16–22 June 2024; pp. 16965–16974. [CrossRef]
21. Zong, Z.; Song, G.; Liu, Y. DETRs with collaborative hybrid assignments training. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), Paris, France, 1–6 October 2023; pp. 6748–6758. [CrossRef]
22. OpenMMLab. (n.d.). MMDetection: RetinaNet [Software Repository]. GitHub. Available online: <https://github.com/open-mmlab/mmdetection/tree/main/configs/retinanet> (accessed on 21 November 2025).
23. Peterande. (n.d.). D-FINE [Software Repository]. GitHub. Available online: <https://github.com/Peterande/D-FINE> (accessed on 21 November 2025).
24. OpenMMLab. (n.d.). MMDetection: DETR [Software Repository]. GitHub. Available online: <https://github.com/open-mmlab/mmdetection/tree/main/configs/detr> (accessed on 21 November 2025).
25. OpenMMLab. (n.d.). MMDetection: DINO [Software Repository]. GitHub. Available online: <https://github.com/open-mmlab/mmdetection/tree/main/configs/dino> (accessed on 21 November 2025).
26. flytocc. (n.d.). MMDetection (Branch Rtdetr): RT-DETR [Software Repository]. GitHub. Available online: <https://github.com/flytocc/mmdetection/tree/rtdetr/configs/rtdetr> (accessed on 21 November 2025).
27. OpenMMLab. (n.d.). MMDetection Projects: CO-DETR [Software Repository]. GitHub. Available online: <https://github.com/open-mmlab/mmdetection/tree/main/projects/CO-DETR> (accessed on 21 November 2025).

28. Ultralytics. (n.d.). YOLOv11 Models [Software Documentation]. Ultralytics Docs. Available online: <https://docs.ultralytics.com/models/yolo11/> (accessed on 21 November 2025).
29. Ivashchuk, O.; Atymtayeva, L.; Zhigalov, A.; Yagalieva, B.; Ivashchuk, O.; Fedorov, V. Neural network methods for the detection of farm animals in dense dynamic groups on images. *Appl. Math. Inf. Sci.* **2024**, *18*, 241–249. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.