Revista de Informática Teórica e Aplicada - RITA - ISSN 2175-2745
Vol. 33, Num. 2 (2026) 154-160

RESEARCH ARTICLE

Poultry Detection on Hooks in Meat Processing Plant Production Lines

Detecção de Aves em Ganchos em Linhas de Produção de Plantas de Processamento de Carne

Leonardo dos Santos Valejo[1], Marlon Marcon[1], André Roberto Ortoncelli[1]*

Abstract: This study focuses on automating chicken production monitoring in a meatpacking plant, applying YOLO to detect birds on hooks on production lines. We detect two objects in authentic images: a chicken (birds hanging on the hook) and a hook (the metal support structure). By counting these objects in videos, we can monitor daily production and identify faults on the production line. We annotated 400 images for the experiment and used 320 to train seven YOLO versions, including lightweight variants (nano, tiny, and small) optimized for low-end devices. In testing, YOLOv9-tiny achieved 0.9967 precision but had lower recall than other experimental instances. YOLOv11-nano achieved an F1-score of 0.9924 and a mAP@50 of 0.9947, standing out in detection quality. Despite minor differences, the results were close for the YOLO versions analyzed, demonstrating a considerable ability to predict the analyzed objects.
Keywords: poultry detection — object detection — deep learning — YOLO

Resumo: Este estudo se concentra na automação do monitoramento da produção de frango em um frigorífico, aplicando o YOLO para detectar aves em ganchos em linhas de produção. Detectamos duas classes de objetos em imagens autênticas: frango (aves penduradas no gancho) e gancho (a estrutura metálica de suporte). Ao contar esses objetos em vídeos, podemos monitorar a produção diária e identificar falhas na linha de produção. Anotamos 400 imagens para o experimento e usamos 320 delas para treinar sete versões do YOLO, incluindo variantes leves (nano, tiny e small) otimizadas para dispositivos de baixo processamento. Nos testes, o YOLOv9-tiny atingiu 0.9967 de precisão, porém obteve um recall inferior a outras instancias experimentais. Já o YOLOv11-nano alcançou um F1-score de 0.9924 e um mAP@50 de 0.9947, se destancado em termos de qualidade da detecção. Apesar de pequenas diferenças, os resultados foram próximos para as versões do YOLO analisadas, demonstrando uma capacidade considerável de prever os objetos analisados.
Palavras-Chave: detecção de aves — detecção de objetos — aprendizado profundo — YOLO

[1] *Universidade Tecnológica Federal do Paraná (UTFPR), Brazil*
***Corresponding author**: ortoncelli@utfpr.edu.br
DOI: http://dx.doi.org/10.22456/2175-2745.150924 • **Received:** 14/10/2025 • **Accepted:** 03/11/2025

1. Introduction

The chicken processing industry is one of the most advanced in mechanical automation, handling large production volumes under strict health and quality standards. However, many activities, such as counting contamination and fractures, are still commonly performed with visual inspection, making the process susceptible to errors [1].

Computer vision methods based on deep learning have been successfully applied in various stages of the chicken production process on farms and meat processing plant production lines. Despite the volume of applications developed, there is room for improvement and new solutions related to the meatpacking plants' production cycle [2, 3, 4, 5, 6].

This study presents preliminary results from a project aimed at automating chicken production monitoring in a unit that slaughters approximately 12 thousand poultry hour. This project aims to demonstrate that modern object detection methods based on deep learning can accurately detect two classes of objects: chicken and hook. The ”chicken” label refers to the housed and plucked animal hanging from a hook. The ”hook” class refers to the metal structure that carries the chickens. Counting these two classes in videos will allow us to measure the number of chickens processed daily and identify production slots without chickens (hooks with a chicken directly below), as shown in Figure 1.

Detecting poultry in meat processing plants production

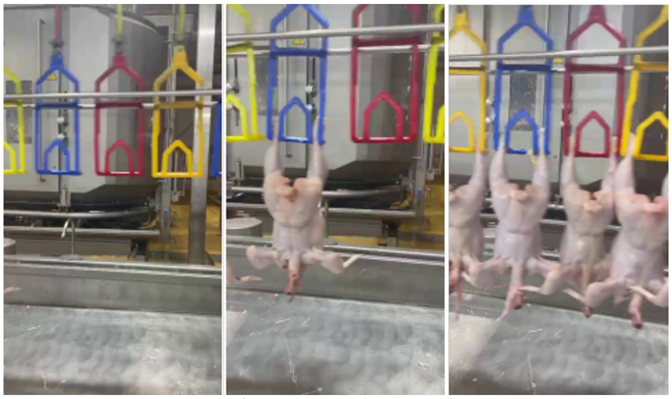

Figure 1. Examples of annotated images from the dataset showing chickens and hooks on the production line

lines is important because: i) it is a preliminary step for other Artificial Intelligence applications to improve tasks and automate manual tasks; ii) it allows the generation of accurate reports counting the number of chickens that passed through each stage of the production line; and iii) it allows for the rapid identification of problems on the production line, issuing alerts quickly. Regarding alerts, chickens can fall to the ground at different points due to the high speed of the slaughter line. According to the law, if they fall to the ground, the chicken becomes unfit for human consumption and must be discarded or used for animal feed production, generating waste and reduced revenue.

In our experiment, we recorded a video of chickens hanging from overhead hooks. We manually annotated the objects of interest in 400 selected frames, each containing different instances of chickens. We used 320 frames to train models based on the You Only Look Once (YOLO) architecture, one of the most popular families for object detection in images and videos [7].

We trained seven versions of YOLO (v5-Nano, v8-Nano, v9-Tiny, v10-Nano, v11-Nano, v12-Nano, and v12-Small). Since we aim to deploy the developed solution on edge devices in the future, we chose the nano, tiny, and small variants designed explicitly for hardware with limited computing resources, such as cell phones, drones, and embedded systems [8]. We obtained an average mean average precision (mAP) of 0.9942 for each YOLO version analyzed, considering an Intersection of Union (IoU) of 0.5. The best result in terms of mAP@50 was 0.9948, obtained with YOLOv11-nano. Regarding precision and F1-score, the best results were 0.9967 (99,67%) and 0.9924 with the YOLOv9-tiny and YOLOv1-nano versions of YOLO, respectively. Despite these minor differences, the results across the evaluated YOLO versions were consistently close, demonstrating a robust ability to detect the analyzed objects. This indicates that even lightweight YOLO models are suitable for accurate and efficient deployment in real-world scenarios involving resource-constrained devices.

The remainder of this work is organized as follows. Theoretical aspects for understanding the project and related works are in Section 2.3. The experimental methodology is in Section 3. The results obtained are in Section 4. Finally, conclusions and proposals for future work are in Section 5.

2. Related Works and Foundations

2.1 Poultry production line

This Subsection aims to present a roadmap of the entire process that poultry follows in the meat processing plant production lines (considering the partner company of this project), aiming to contextualize the moment of application of the detection methods presented in this work and highlighting their importance in different stages of the process.

The chicken arrives at the slaughterhouse alive, is manually hung upside down on the overhead conveyor, and is then stunned by passing through a blue-lit environment and an electrically charged water bath with controlled current. An automatic machine then cuts the neck, and if the cut is not sufficient, a worker completes the bleeding manually. The bird remains in the bleeding area for about 90–120 seconds.

Each chicken then goes through the following steps: i) scalding, where it is immersed in hot water to loosen the feathers; ii) mechanical defeathering, performed by automated machines with rotating rubber fingers; iii) feces removal, by suction machines; iv) cloaca extraction, performed by a dedicated machine (if the removal is not well executed, bile and feed can contaminate the carcass); v) abdominal opening; vi) washing of the carcass and evisceration, where machines remove the viscera (heart, liver, etc.); vii) inspection by the Federal Inspection Service (SIF) of viscera and carcasses; viii) removal of lungs, crop, and trachea by automatic equipment; ix) inspection at critical control points (PCC); x) rewashing of the carcass; xi) automatic unhooking from the hook; xii) prechilling, where the carcass is immersed in large tanks of cold, chlorinated water for about one hour; xiii) rehanging, where workers manually place carcasses back on the line to drain; xiv) finally, the chicken is sent to the cutting room, where it is portioned and directed to different processing lines.

We trained the chicken detector proposed in this work with images collected immediately after the removal of the crop and trachea. Nevertheless, we intend to extend the approach to monitor all processing stages.

2.2 Object detection

Over the past two decades, object detection has established itself as one of the main challenges in computer vision, receiving significant attention in the field. The development of this technique has advanced considerably, primarily driven by the use of Deep Learning-based methods [9]. Generally, advances have focused on two main approaches: i) two-stage detectors, such as the R-CNN family [10] and its variations [11]; and ii) single-stage detectors, exemplified by YOLO [12] and its more recent versions [13].

Two-stage algorithms generally divide the detection task into two distinct stages: proposition and classification. Single-stage algorithms, on the other hand, perform object detection and classification in a single pass through the image. Each strategy has strengths and limitations: single-step methods tend to be faster but more difficult to handle small or partially occluded objects. On the other hand, two-step methods tend to achieve higher accuracy, although they require more processing time due to the intermediate region-proposition step [14].

Researchers have successfully applied YOLO variations in the literature, achieving high accuracy rates in detecting different objects in images. These applications include bird/chicken detection [15, 16] and the recognition of other animals, such as bees [17] and cats [18]. They have also explored YOLO in other domains, including plant detection [19], defect identification in production lines [20], and medical applications [21].

The YOLO framework, developed by Joseph Redmon, revolutionized real-time object detection by introducing a grid-based approach to predict bounding boxes and class probabilities simultaneously [12]. Since its inception, the YOLO architecture has continuously evolved, with each version, from YOLOv1 to the most recent, introducing accuracy, speed, and efficiency improvements. Significant advancements include introducing anchor boxes, which improve the model's ability to detect objects of varying shapes and sizes, and new loss functions to optimize performance. Additionally, each YOLO iteration features optimized backbones that contribute to faster processing times and better detection capabilities [7].

In this work we explore six YOLO versions: i) YOLOv5 [22]; ii) YOLOv8 [23]; ii) YOLOv9 [24]; iv) YOLOv10 [25]; v) YOLOv11 [26]; e vi) YOLOv12 [27].

In addition to the versions, YOLO also has architectural variations within the same version, adjusted to balance performance and computational cost, such as the tiny, small, and nano versions. The nano versions are aimed at extremely hardware-constrained devices, with top priority given to speed and low memory consumption. The tiny versions are designed to be lightweight compared to the whole model, maintaining a reasonable balance between speed and accuracy. The small versions are intermediate between the tiny and the standard model, offering more layers and parameters than the tiny ones, with gains in accuracy, while maintaining a reduced computational cost. These variations make YOLO a flexible family of models that are adaptable to mobile and embedded device applications and systems with higher processing power [8]. We use nano, small, and tiny variations of the selected YOLO versions.

2.3 Artificial Intelligence in the poultry industry

In recent years, the poultry industry has made significant progress with adopting automation, driven by various technological innovations such as monitoring tools, IoT devices, and smart sensors. In this context, Artificial Intelligence (AI) has become crucial in various sectors, including poultry production. Machine Learning and Deep Learning technologies, subfields of AI, have shown promise in addressing the sector's challenges, enabling data-driven agriculture and optimizing various task management systems in the production chain. Recent literature reviews highlight computer vision-based methods used to support chicken farming [28, 29, 2].

In addition to AI methods applied to chicken farms, this technique has also been applied to support meat processing plant production lines. In this section, we present details of some of these recent applications directly related to our project.

On a production line, the process begins with electrical stunning of the animal, which is performed before slaughter. In [3], an algorithm is presented that recognizes three stunning conditions: insufficient stunning, moderate stunning, and excessive stunning. A model was developed based on Convolutional Neural Networks (CNNs) to detect stunning broilers' conditions. When applied to a boosted dataset containing 27,828 images of stunned chickens, the model's identification accuracy was 98.06%.

With the chicken already placed on hooks on the production line, the work of [4] stands out, focusing on recognizing wing breakage and/or disarticulation. The authors used color and morphology segmentation techniques to detect and count the birds, achieving promising results (97.67% accuracy). Although the method is effective in controlled contexts, its approach relies on manual adjustments. It is sensitive to variations in lighting and positioning, which limits its scalability in more complex industrial environments.

Also, for slaughtered chickens hanging on hooks, in [30], a hyperspectral-multispectral line-scan imaging system was developed to differentiate healthy and systemically diseased chickens. Tests were conducted on chickens in a line moving at a speed of 70 birds per minute. The calibration dataset was used to develop the imaging system parameters for multispectral inspection based on fuzzy logic detection algorithms using selected key wavelengths. The classification achieved 97.6% accuracy for healthy birds and 96% for systemically diseased birds.

In poultry slaughter, accurate positioning of viscera is critical to avoid damage. [5] presents an improved region-based active contouring method with a level-set formulation. This method, combined with several color space conversion and top-low-hat transformation operations, accurately extracted the viscera contour and removed noise. The results showed that the recognition accuracy of the heart-liver and fat areas in the viscera was 98.98% and 99.75%, respectively, while the overall accuracy for poultry viscera was 98.96%.

It is worth highlighting that detecting chickens and/or parts of chickens slaughtered on production lines is an important step for the methods mentioned and for new methods that can be developed, considering all the production stages described in Subsection 2.1.

3. Methodology

In a real meat processing plant production line, we recorded a video of plucked chickens passing through hooks during the production process. We recorded the production line with a smartphone camera. From the video, we extracted 400 frames (ensuring distinct chickens in each image), which were manually annotated into two classes (chicken and hook) using the MakeSense.ai platform[1].

We randomly selected 320 images (80%) for the training set and allocated the remaining 80 images (20%) to the validation set. As for the objects annotated in each image, in the training set, there are 1381 instances of chicken and 604 of hook, while in the validation set, the chickens appear 325 times, and the hooks 295 times. We resized all images to 640 × 640 pixels and applied several data augmentation techniques to the training set, including mirroring, brightness variations, and scaling. We exported the annotations in YOLO format with normalized coordinates.

We conducted experiments with seven different versions/ variations of YOLO: i) YOLOv5-Nano; ii) YOLOv8-Nano; iii) YOLOv9-Tiny; iv) YOLOv10-Nano; v) YOLOv11-Nano; vi) YOLOv12-Nano; and vii) YOLOv12-Small.

For both experiments, the following hyperparameters were configured: the batch size was fixed at 5, while the other settings remained at the Ultralytics YOLO framework's default values, including the Stochastic Gradient Descent (SGD) optimizer and the initial learning rate of 0.01. During training, performance was monitored by evaluating validation metrics, learning curves, and periodic weight saving.

We conducted the experiments on a Dell notebook equipped with Intel® UHD Graphics (128 MB VRAM), 8 GB RAM, and a 12th Gen Intel® Core™ i5-1235U processor.

Considering that Intersection over Union (IoU) is a metric used to evaluate how well a box predicted by the detector overlaps with the ground truth in object detection tasks, we computed the following experimental metrics to evaluate the experiments:

- mAP@50: Measures the average precision (AP), considering that a prediction is correct if the IoU is greater than or equal to 0.50.
- mAP@50-95: Measures the AP at 10 different IoU thresholds (0.50, 0.55, 0.60, ..., up to 0.95). This metric is more difficult to achieve high values, but it better reflects the model's overall quality in terms of accurate object localization.
- Precision: Measures the proportion of correct detections relative to the total number of detections made by the model. High precision indicates that the model makes few false positives.
- Recall: Measures the proportion of correctly detected objects relative to the total number of objects present in the images. High recall indicates that the model misses few true objects (false negatives).
- F1-score: The harmonic mean between precision and recall, providing a balanced metric that considers both false positives and false negatives.
- FPS: Indicates the number of frames the model could process per second during the tests performed. FPS was computed when the trained model was applied to object detection in the validation set, on the same computer used for training.

4. Results

Table 1 presents the experimental metrics computed for each trained version of YOLO applied to the validation set. Figure 2 presents a confusion matrix of the results obtained with the trained model that had the best accuracy. Figure 3 presents an example of object detection, also for the best model, considering the same images presented in Figure 1.

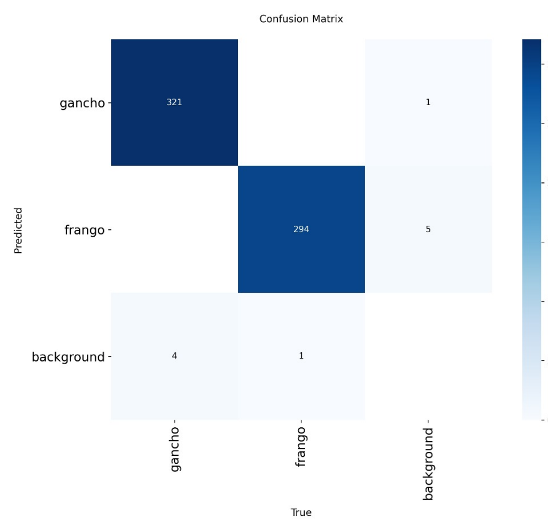

Figure 2. Confusion matrix of the best trained model

In the experiments, six versions of YOLO achieved an F1-score higher than 0.96, with the best F1-score (0.9924) obtained with YOLOv11-nano. In terms of mAP@50, all the models presented results above 0.99, with the best results (0.9948) also obtained with YOLOv11-nano. In this context, YOLOv11-nano stands out for its best detection results. Although YOLOv9-tiny has 0.9967 of precision, it has considerably lower recall, which may preclude its use.

Although it excelled in detection quality in the experiment, YOLOv9-tiny performed considerably worse in FPS, being approximately 40% slower than YOLOv11-nano. Considering both FPS, YOLOv11-nano may represent a more balanced choice.

To visually analyze the relationship between FPS and the F1-score and mAP@50 values obtained for each model,

[1] https://www.makesense.ai/

Model	Precision	Recall	F1-score	mAP@50	mAP@50-95	FPS
YOLOv5-nano	0.9933	0.9886	0.9909	0.9948	0.7849	11.98
YOLOv8-nano	0.9900	0.9905	0.9902	0.9948	0.7826	11.87
YOLOv9-tiny	0.9967	0.9791	0.9879	0.9946	0.7270	7.07
YOLOv10-nano	0.9637	0.9607	0.9622	0.9906	0.7890	10.55
YOLOv11-nano	0.9915	0.9934	0.9924	0.9948	0.7820	11.73
YOLOv12-nano	0.9892	0.9915	0.9903	0.9948	0.7771	9.32
YOLOv12-small	0.9890	0.9943	0.9916	0.9948	0.7855	5.14

Table 1. Experimental results obtained with the different versions of YOLO used

we created two graphs in Figures 4 and 5. In Figure 4, the x and y-axes represent the mAP@50 x FPS rates for each trained model. Figure 5 is similar, but presents the F1-score obtained instead of the mAP@50. We believe that, despite the good results obtained in terms of detection quality, in future work we will need to adapt these models for edge deployment, such as network pruning, weight quantization, and other compression techniques [31, 32].

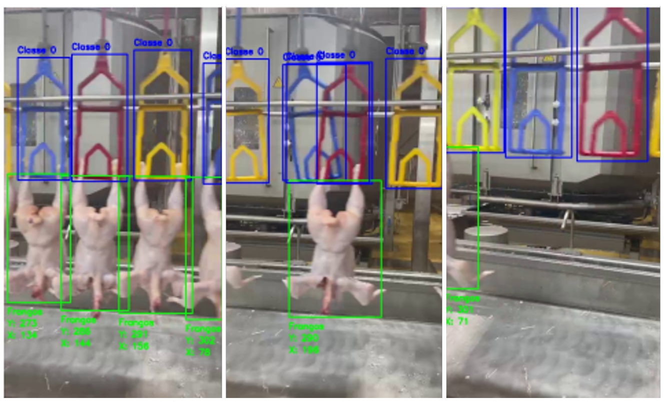

Figure 3. Objects detected with the best trained model

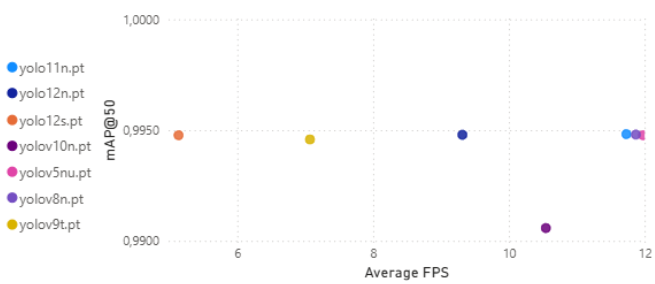

Figure 4. Relation of mAP@50 x FPS for all YOLO versions evaluated

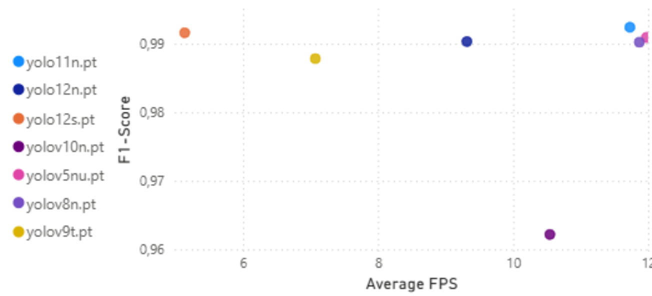

Figure 5. Relation of F1-score x FPS for all YOLO versions evaluated

5. Conclusion

This study presented preliminary results on automating the monitoring of poultry production lines using YOLO-based object detection models. We focused on directly detecting two relevant classes—chickens and hooks—on a real slaughterhouse production line. The experimental results demonstrated that even lightweight YOLO variants, such as Nano and Tiny, achieved high precision, recall, and F1-scores, confirming the feasibility of the proposed approach.

This study demonstrated the feasibility of applying lightweight YOLO models for detecting chickens and hooks on poultry processing plant production lines. The experimental results showed that even compact architectures such as nano, tiny, and small versions achieved high performance, with F1-scores above 0.96 and mAP@50 values close to 0.99 in most cases. These results confirm the robustness of YOLO models in distinguishing objects of interest under real industrial conditions, even when trained with a relatively small dataset.

Although YOLOv11-nano achieved slightly higher detection metrics (F1 = 0.9924, mAP@50 = 0.9948), YOLOv5-nano delivered nearly identical accuracy while reaching the highest inference speed (11.98 FPS). Therefore, YOLOv5-nano stands out as the most efficient option, offering the best balance between detection performance and processing speed for real-time deployment.

Despite the promising results, the experiments were conducted on a computer that does not represent the computational limitations typically found in embedded or low-cost devices. As future work, we intend to adapt the models for edge deployment through techniques such as network pruning, weight quantization, and knowledge distillation, enabling real-time operation with minimal accuracy loss.

Additionally, the detection system could be extended to other stages of the poultry processing chain. For example, it could automatically verify the effectiveness of stunning and bleeding before scalding, helping to prevent animal mistreatment, ensure regulatory compliance, and minimize product waste. Such developments would further enhance the system's contribution to automation and quality control in meat processing plants.

Author Contributions

Leonardo dos Santos Valejo was responsible for producing and annotating the image dataset and for executing all experimental procedures. Marlon Marcon and André Roberto Ortoncelli supervised the research, guided the methodological decisions, and validated the experimental results. All authors contributed equally to the writing, revision, and final approval of the manuscript.

References

[1] BASSI, N. S. S.; SILVA, C. L. Oportunidades e desafios em pd&i na cadeia produtiva de frangos de corte. *Agropensa, Embrapa Suínos e Aves. Universidade Tecnológica Federal do Paraná (UTFPR)*, 2017.

[2] UMURUNGI, S. N. et al. Leveraging the potential of convolutional neural networks in poultry farming: a 5-year overview. *World's Poultry Science Journal*, Taylor & Francis, v. 81, n. 1, p. 3–34, 2025.

[3] YE, C.-w. et al. Broiler stunned state detection based on an improved fast region-based convolutional neural network algorithm. *Poultry science*, Elsevier, v. 99, n. 1, p. 637–646, 2020.

[4] SILVA, H. R. D.; WEIZENMAN, G. D. S.; MUTZENBERG, R. Algoritmo para reconhecimento automatizado de defeitos em frangos de linha de produção utilizando visão computacional. In: *Congresso Brasileiro de Automática-CBA*. [S.l.: s.n.], 2019. v. 1, n. 1.

[5] CHEN, Y. et al. Machine vision on the positioning accuracy evaluation of poultry viscera in the automatic evisceration robot system. *International Journal of Food Properties*, Taylor & Francis, v. 24, n. 1, p. 933–943, 2021.

[6] JIAN, R. et al. Nondestructive freshness recognition of chicken breast meat based on deep learning. *Scientific Reports*, Nature Publishing Group UK London, v. 15, n. 1, p. 27538, 2025.

[7] ALI, M. L.; ZHANG, Z. The yolo framework: A comprehensive review of evolution, applications, and benchmarks in object detection. *Computers*, MDPI, v. 13, n. 12, p. 336, 2024.

[8] WONG, A. et al. Yolo nano: A highly compact you only look once convolutional neural network for object detection. In: IEEE. *Fifth Workshop on Energy Efficient Machine Learning and Cognitive Computing-NeurIPS Edition (EMC2-NIPS)*. [S.l.], 2019. p. 22–25.

[9] MOHAMMED, S. Y. Architecture review: Two-stage and one-stage object detection. *Franklin Open*, Elsevier, p. 100322, 2025.

[10] GIRSHICK, R. et al. Rich feature hierarchies for accurate object detection and semantic segmentation. In: *IEEE conference on computer vision and pattern recognition*. [S.l.: s.n.], 2014. p. 580–587.

[11] PRIYADHARSHINI, G.; DOLLY, D. R. J. Comparative investigations on tomato leaf disease detection and classification using cnn, r-cnn, fast r-cnn and faster r-cnn. In: IEEE. *International conference on advanced computing and communication systems (ICACCS)*. [S.l.], 2023. v. 1, p. 1540–1545.

[12] REDMON, J. et al. *You only look once: Unified, real-time object detection*. 2016. 779–788 p.

[13] HUA, Z. et al. A benchmark review of yolo algorithm developments for object detection. *IEEE Access*, IEEE, 2025.

[14] KOIRALA, A. et al. Deep learning–method overview and review of use for fruit detection and yield estimation. *Computers and electronics in agriculture*, Elsevier, v. 162, p. 219–234, 2019.

[15] REDDY, D. J. et al. Wild bird detection on airborne imagery using modified yolo network. In: SPRINGER. *International Conference on Information and Management Engineering*. [S.l.], 2023. p. 9–23.

[16] JIANG, D. et al. Real-time tracker of chicken for poultry based on attention mechanism-enhanced yolo-chicken algorithm. *Computers and Electronics in Agriculture*, Elsevier, v. 237, p. 110640, 2025.

[17] LIMA, I. C. R. et al. Analyzing bee behavior through video tracking using computer vision techniques. In: SBC. *Workshop de Visão Computacional (WVC)*. [S.l.], 2024. p. 280–286.

[18] LEMOS, D. V. X. et al. Yolo and cnn for cat detection and recognition. *Revista de Informática Teórica e Aplicada*, v. 32, n. 1, p. 158–165, 2025.

[19] MARTINS, T. M. et al. Detection of cacao trees in orthomosaic images using yolov8 and sahi. In: SBC. *Workshop de Visão Computacional (WVC)*. [S.l.], 2024. p. 37–41.

[20] SEIDEL, R.; WAIANDT, C. A.; JÚNIOR, H. S. Defect detection in textile manufacturing: Improving findings using yolov5 versions. In: SBC. *Workshop de Visão Computacional (WVC)*. [S.l.], 2023. p. 66–71.

[21] RAGAB, M. G. et al. A comprehensive systematic review of yolo for medical object detection (2018 to 2023). *IEEE Access*, IEEE, v. 12, p. 57815–57836, 2024.

[22] HUSSAIN, M. Yolov5, yolov8 and yolov10: The go-to detectors for real-time vision. *arXiv preprint arXiv:2407.02988*, 2024.

[23] SOHAN, M.; RAM, T. S.; REDDY, C. V. R. A review on yolov8 and its advancements. In: SPRINGER. *International Conference on Data Intelligence and Cognitive Informatics*. [S.l.], 2024. p. 529–545.

[24] YASEEN, M. What is yolov9: An in-depth exploration of the internal features of the next-generation object detector. *arXiv preprint arXiv:2409.07813*, 2024.

[25] WANG, A. et al. Yolov10: Real-time end-to-end object detection. *Advances in Neural Information Processing Systems*, v. 37, p. 107984–108011, 2024.

[26] KHANAM, R.; HUSSAIN, M. Yolov11: An overview of the key architectural enhancements. *arXiv preprint arXiv:2410.17725*, 2024.

[27] TIAN, Y.; YE, Q.; DOERMANN, D. Yolov12: Attention-centric real-time object detectors. *arXiv preprint arXiv:2502.12524*, 2025.

[28] SUBRAMANI, T.; JEGANATHAN, V.; BALASUBRAMANIAN, S. K. Machine learning and deep learning techniques for poultry tasks management: a review. *Multimedia Tools and Applications*, Springer, v. 84, n. 2, p. 603–645, 2025.

[29] AZIZ, N. S. N. A. et al. A review on computer vision technology for monitoring poultry farm—application, hardware, and software. *IEEE access*, IEEE, v. 9, p. 12431–12445, 2020.

[30] CHAO, K. et al. Hyperspectral-multispectral line-scan imaging system for automated poultry carcass inspection applications for food safety. *Poultry science*, Elsevier, v. 86, n. 11, p. 2450–2460, 2007.

[31] LI, Z. et al. Channel pruned yolov5-based deep learning approach for rapid and accurate outdoor obstacles detection. *arXiv preprint arXiv:2204.13699*, 2022.

[32] LYU, Z. et al. A survey of model compression strategies for object detection. *Multimedia tools and applications*, Springer, v. 83, n. 16, p. 48165–48236, 2024.