EAI Endorsed Transactions
on Industrial Network and Intelligent Systems Research Article EAI.EU

Integrating YOLOv8-agri and DeepSORT for Advanced Motion Detection in Agriculture and Fisheries

Hieu Duong-Trung[1,*], Nghia Duong-Trung[2]

[1]Can Tho University, 3/2 street, Xuan Khanh ward, Ninh Kieu district, Can Tho city, Vietnam
[2]German Research Center for Artificial Intelligence (DFKI), Alt-Moabit 91 C, 10559, Berlin, Germany

Abstract

This paper integrates the YOLOv8-agri models with the DeepSORT algorithm to advance object detection and tracking in the agricultural and fisheries sectors. We address the current limitations in object classification by adapting YOLOv8 to the unique demands of these environments, where misclassification can hinder operational efficiency. Through the strategic use of transfer learning on specialized datasets, our study refines the YOLOv8-agri models for precise recognition and categorization of diverse biological entities. Coupling these models with DeepSORT significantly enhances motion tracking, leading to more accurate and reliable monitoring systems. The research outcomes identify the YOLOv8l-agri model as the optimal solution for balancing detection accuracy with training time, making it highly suitable for precision agriculture and fisheries applications. We have publicly made our experimental datasets and trained models publicly available to foster reproducibility and further research. This initiative marks a step forward in applying sophisticated computer vision techniques to real-world agricultural and fisheries management.

Received on 15 December 2023; accepted on 11 February 2024; published on 12 February 2024

Keywords: YOLOv8; DeepSort; Motion Detection; Agricultural Datasets; Reproducibility; Open Data

doi:10.4108/eetinis.v11i1.4618

1. Introduction

The current scope of computer vision technology marks a revolutionary leap in how machines interact with and interpret visual information. This rapid evolution is largely fueled by significant advancements in deep learning and artificial intelligence, which have enabled machines to execute tasks like image classification, object detection, and semantic segmentation with unprecedented accuracy and efficiency [1, 2]. These advances have pushed the boundaries of computer vision far beyond its traditional academic and industrial confines, paving the way for its integration into a wider array of sectors. The impact of these state-of-the-art algorithms is particularly evident in their ability to handle complex tasks, adapt to various environments, and process vast quantities of visual data, thus opening new frontiers in technological applications.

In the context of agriculture and fisheries, the role of computer vision has been transformative, propelling these sectors into a new era of technological innovation. In agriculture, computer vision technologies facilitate advanced farming techniques, enabling enhanced monitoring of crop health, efficient pest management, and optimized resource allocation. These capabilities are crucial for improving crop yields, reducing waste, and ensuring sustainable farming practices [3–5]. In the fisheries sector, computer vision plays a vital role in the sustainable management of marine resources. It assists in monitoring fish populations, assessing the health of marine ecosystems, and managing fish sorting and processing. The application of computer vision in these sectors extends beyond mere object identification; it involves the interpretation of complex scenes and activities, leveraging the technology's ability to rapidly analyze and process large volumes of visual data. This not only aids in enhancing productivity but also plays a crucial role in promoting sustainable practices [6–8].

*Corresponding author. Email: hieum3721002@gstudent.ctu.edu.vn

However, computer vision applications, particularly in motion detection within agriculture and fisheries, are fraught with challenges unique to these sectors. These environments are dynamic and unstructured, marked by natural variability and various species, all operating under fluctuating conditions. Effective motion detection in such settings requires adaptive and robust algorithms to variations in lighting, weather, and background activity. The need for real-time processing and analysis is paramount, especially for tasks that require immediate response, such as monitoring animal behavior, detecting signs of disease, or adapting to environmental changes. The complexity of these tasks is compounded by the unpredictable nature of these environments, necessitating a high degree of precision and reliability in the computer vision algorithms employed.

Applying advanced computer vision models like YOLOv8 [9] in these specialized sectors is challenging. These models, often developed and pre-trained on standard datasets, may not seamlessly adapt to the unique requirements of agricultural or marine environments. Despite its advanced object detection capabilities, this limitation becomes apparent when YOLOv8 struggles with accurate classification in these specialized settings. For example, the model might inaccurately categorize fish and chickens as generic birds due to the absence of specific training on relevant datasets. We present the detection performance of the public pre-trained Yolov8 models in Figures 1 and 2. Note that pre-trained YOLO models, available to the public, have typically undergone extensive training on large and diverse datasets, such as COCO (Common Objects in Context) and ImageNet. These datasets contain millions of labeled images covering a wide array of categories. However, those public models made wrong predictions in several scenarios, raising awareness of the training datasets' robustness, transferability, and representation of the training datasets [10, 11]. One of the main reasons is that the target classes in a specific domain do not exist in the original training datasets. This misclassification highlights a critical gap in applying general-purpose computer vision models to specialized domains. It underscores the need for customizing and fine-tuning these models with domain-specific data, ensuring they are trained to recognize and understand the unique aspects of these environments. While the advancements in computer vision, as exemplified by models like YOLOv8 [12, 13], hold immense promise, their effective application in sectors such as agriculture and fisheries requires a more nuanced approach. This involves customizing the models to accurately recognize and interpret the specific characteristics of these environments and ensuring that they are trained on appropriate, context-specific datasets. Addressing these challenges is essential for fully harnessing the potential of computer vision in enhancing the efficiency, productivity, and sustainability of these vital sectors. The future of these technologies in agriculture and fisheries hinges on our ability to adapt and tailor them to meet the unique demands and challenges of these fields.

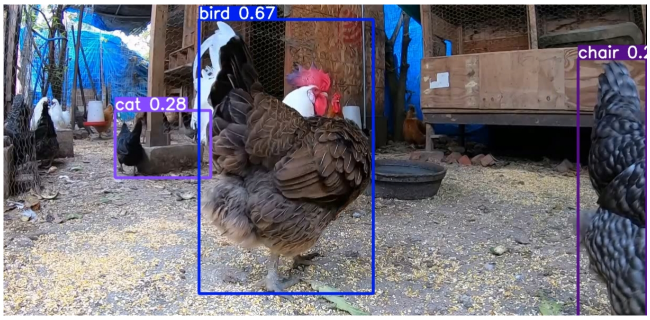

Figure 1. The detection performance of the public pre-trained Yolov8 models on the screenshot of a chicken farming.

Figure 2. The detection performance of the public pre-trained Yolov8 models on the screenshot of a fish farming.

2. Related Work

In recent years, the convergence of deep learning and computer vision has brought groundbreaking changes in diverse fields, notably agriculture and fisheries. A series of research papers have emerged, showcasing these technologies' extensive applications and transformative impact in these sectors.

A survey conducted by Saleh *et al.* focuses on applying deep learning in classifying fish species in underwater habitats, a crucial aspect for the sustainable development of fisheries and environmental conservation [7]. The study delves into the intricate details of how deep learning can effectively differentiate between various fish species, highlighting its potential in marine biology and fishery management. The survey reveals many papers on fish classification using deep learning, indicating the field's growing prominence and the technology's capability to address complex biological classification problems.

The paper discusses the transformative impact of artificial intelligence, particularly deep learning-based computer vision, on modern agricultural practices. It illustrates how these advanced technologies enable automated and more efficient agricultural activities, signifying a pivotal shift in the industry's approach to farming. By integrating deep learning into computer vision systems, a wide array of agricultural tasks, from crop monitoring to pest control, are being automated, enhancing productivity and sustainability in agriculture [3].

Mei *et al.* reviews recent progress in applying deep learning algorithms for tracking fish behavior, a critical challenge in aquaculture. The study highlights the innovations brought about by deep learning in visual tracking, particularly in understanding and managing fish behavior. This emerging field of research demonstrates the capabilities of deep learning to provide detailed insights into marine life, aiding in more scientific management and potentially reducing losses due to disease and other environmental factors [14].

Coulibaly *et al.* conducted a bibliometric analysis on more than 400 recent papers to highlight the evolution and advances observed in deep learning in agriculture. The study showcases how deep learning accelerates and improves data processing in agriculture, contributing significantly to the digital transformation of the field. This analysis underlines the growing influence of artificial intelligence in agriculture, particularly in enhancing efficiency and precision in various farming practices [15].

The impact of computer vision and deep learning on agricultural advancement was discussed by Paul *et al.* [16]. The extensive survey discusses the significant role of computer vision, augmented by deep learning, in the agricultural sector, specifically in inspecting and grading food and agricultural products. The paper details these technologies' applications in enhancing agricultural production's quality and efficiency. It provides a comprehensive overview of how machine learning and computer vision are being used to revolutionize traditional farming practices, offering new ways to assess and improve produce quality.

Qin *et al.* explores the integration of YOLO-based models in precision agriculture. This paper focuses on developing an efficient, cost-effective, and real-time system for spraying pesticides using drones. Ag-YOLO employs an onboard computer vision component designed for Unmanned Aerial Vehicles (UAVs). It demonstrates a remarkable balance of high accuracy and speed in detecting crops for targeted pesticide application, highlighting its potential to enhance agricultural practices, especially in challenging terrains and small fields [17]. Another research, Automated Detection, Classification and Counting of Fish in Fish Passages With Deep Learning, is part of Canada's Ocean Aware project and focuses on applying deep learning techniques for automated fish detection, classification, and tracking. It takes a different approach by applying YOLO models to the field of fisheries. This study is part of a larger project to understand and mitigate human impacts on at-risk fish species. It utilizes deep learning algorithms, specifically YOLOv3 and Mask-RCNN, to automatically detect, classify, and track various fish species. Analyzing high-resolution sonar and video data provides valuable insights into fish behavior and movement, offering tools for better fishery management and conservation efforts [18].

The combination of YOLO models and DeepSort algorithms for motion tracking covers a variety of applications, highlighting the versatility and effectiveness of these technologies in different domains. In a study led by Durve *et al.* [19], the efficacy of YOLOv5 and its successor, YOLOv7, is explored to track microfluidic droplets. This research demonstrates the customization and utility of these advanced object detection and tracking algorithms in a highly specialized field, providing insights into the analysis of microfluidic videos for physical quantity inference. Another research effort delves into enhancing object detection and tracking systems, crucial in fields like video analysis, action recognition, and smart elderly care. This study underlines the significance of a robust tracking system that can accurately recognize objects and follow their trajectories, employing spatial data from various sources, including drones and multi-camera setups [20]. Research on pedestrian tracking using YOLOv5 in conjunction with DeepSORT reveals the algorithms' effectiveness in handling challenges like sudden movement, occlusion, and appearance changes in real-time scenarios. These studies highlight the potential of these technologies in autonomous driving and intelligent surveillance, where accurate pedestrian detection and tracking are crucial [21, 22].

Exploring advanced technological integrations in agriculture and fisheries has seen significant progress in recent years. Among the notable developments in this area, the application of AI, mainly through object detection and tracking algorithms, stands out as a promising frontier. Despite this, there remains a conspicuous research gap in the specialized application of cutting-edge AI models, specifically the integration of YOLOv8 and DeepSORT, for general motion-tracking purposes as highlighted in previous studies [23, 24]. While laying the groundwork for understanding the capabilities of such integrations, these studies predominantly focus on urban and controlled environments, leaving a vast unexplored territory in sectors as critical as agriculture and fisheries.

The need for dedicated research into applying the latest YOLOv8 in conjunction with DeepSORT in agriculture and fisheries is particularly glaring. Agriculture and fisheries are industries fraught with unique challenges, including the need for real-time monitoring and tracking of various entities such as livestock, fish, and even agricultural machinery. These sectors demand accurate, efficient, and robust solutions to handle the dynamic and often unpredictable nature of their environments. The advanced object detection capabilities of YOLOv8, when paired with the sophisticated tracking abilities of DeepSORT, offer a compelling solution to these challenges. However, this potential synergy has yet to be sufficiently explored or documented in the existing body of research.

In addressing this gap, our research seeks to pioneer the integration of YOLOv8 with DeepSORT specifically tailored to the needs of agriculture and fisheries. The enhanced accuracy and speed of YOLOv8 for object detection and the improved tracking consistency and efficiency provided by DeepSORT could revolutionize how these industries monitor and manage their operations. Our investigation aims to demonstrate this integration's practical applicability and benefits, marking a significant step forward in deploying AI technologies in agriculture and fisheries.

It makes our study the first of its kind, aiming not just to bridge a significant gap in applying these advanced technologies within these vital sectors, but also to lay the groundwork for future research and development. We are not only exploring the theoretical underpinnings of integrating YOLOv8 and DeepSORT but also engaging in empirical research to validate our hypotheses. By doing so, we aim to establish a comprehensive framework that can guide the deployment of these technologies, addressing the specific challenges faced by the agriculture and fisheries sectors and ultimately contributing to their efficiency, sustainability, and productivity.

3. Technical Background

3.1. Yolov8

YOLOv8 is a new state-of-the-art computer vision model built by Ultralytics [9], the creators of YOLOv5. The YOLOv8 model contains out-of-the-box support for object detection, classification, and segmentation tasks, accessible through a Python package as well as a command line interface. The advancements in YOLOv8 over its predecessor signify notable improvements in the field of computer vision[1]. The architecture of YOLOv8 is presented in Figure 3.

[1] https://blog.roboflow.com/whats-new-in-yolov8/

The introduction of YOLOv8 marks a significant advancement in the field of computer vision, particularly in object detection, thanks to several key improvements over its predecessor, YOLOv5. One of the most notable enhancements is incorporating a new anchor-free detection system. Like earlier YOLO versions, traditional object detection models typically rely on anchor boxes, which are predefined bounding boxes of various shapes and sizes to detect objects. However, the anchor-free system in YOLOv8 eliminates the need for these preset boxes. Instead, it directly predicts the boundaries of objects in an image. This approach can lead to better accuracy and flexibility, enabling the model to detect objects of varying and unconventional shapes more effectively.

Another significant change in YOLOv8 is modifying the convolutional blocks used in the model. Convolutional blocks form the core of Convolutional Neural Networks (CNNs), which are integral in image processing and analysis. By changing these blocks, YOLOv8 alters how it processes and learns from image data. This could result in more efficient learning patterns, enhanced feature extraction, and overall improved performance in key tasks such as object detection and classification. These modifications are crucial, as they can greatly influence the accuracy and efficiency of the model in real-world applications.

The third major improvement in YOLOv8 is the implementation of mosaic augmentation during the training phase. Mosaic augmentation is a technique where four training images are combined into one composite image. This method exposes the model to a more diverse and complex range of scenes during training, enhancing its ability to generalize and perform accurately in various scenarios. Interestingly, YOLOv8 turns off this augmentation before the last 10 training epochs. This strategy allows the model to refine its learning on standard, non-augmented images, potentially leading to better performance when deployed in real-world environments. This blend of extensive training followed by focused fine-tuning could be a key factor in the enhanced capabilities of YOLOv8 compared to its predecessors.

3.2. DeepSort Algorithm

The DeepSort algorithm is an advanced technique in the field of computer vision and object tracking [25]. It extends the original SORT (Simple Online and Real-time Tracking) algorithm [26], incorporating deep learning to enhance tracking accuracy and robustness. DeepSort primarily addresses the limitations of SORT by improving the tracking of objects during occlusions and across different frames. At its core, DeepSort utilizes deep neural networks to extract features from detected objects. These features then calculate

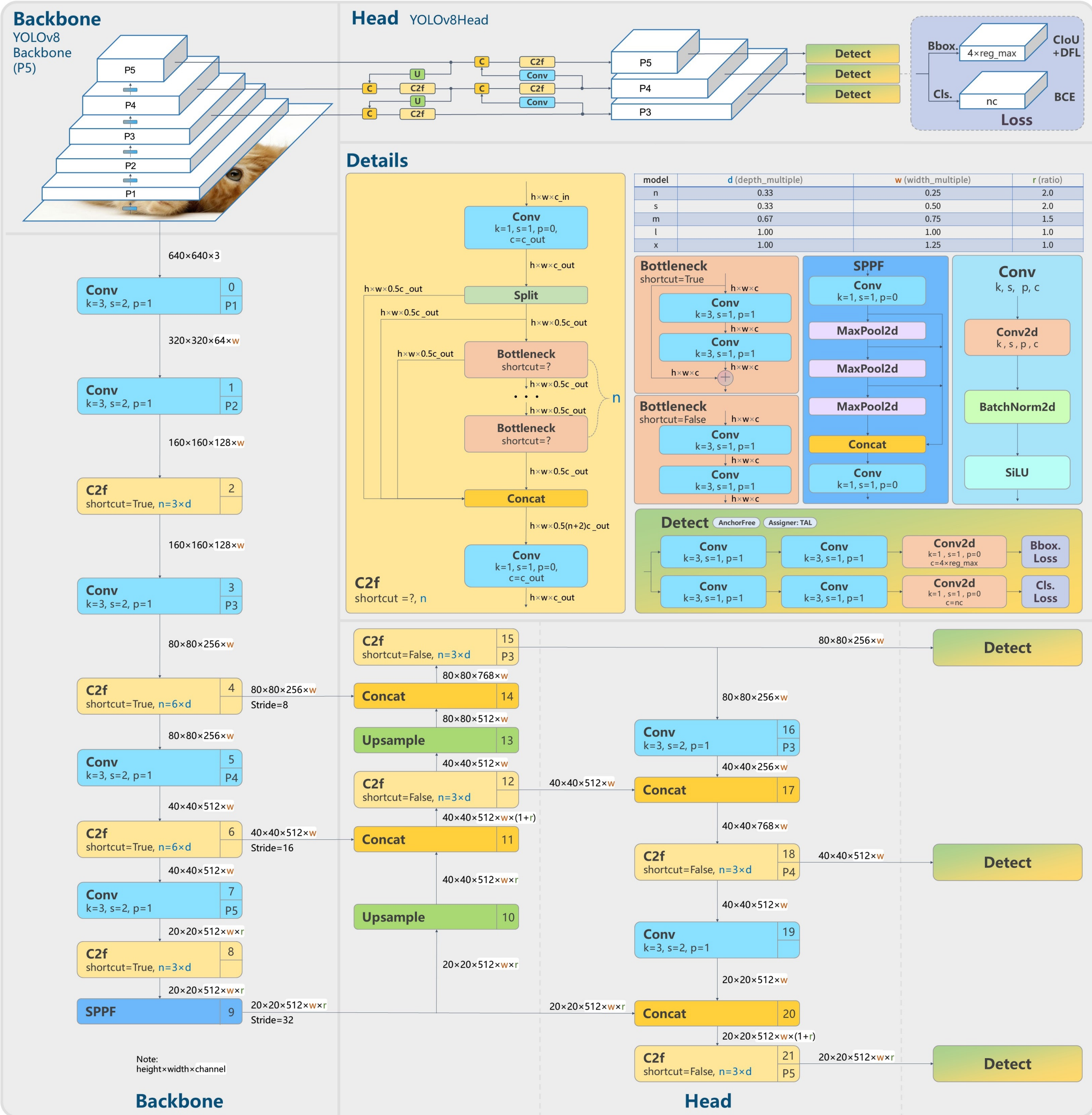

Figure 3. The architecture of YOLOv8. *Image courtesy to Ultralytics and RangeKing at* `https://github.com/ultralytics/ultralytics/issues/189`.

similarity scores between the current and previous detections. This approach significantly improves the ability to track objects over time, even when they might be temporarily occluded or their appearance slightly changes.

The algorithm operates in a multi-step process. Initially, object detections are obtained from each video frame, typically using a separate object detection model like YOLO. Once the objects are detected, DeepSort extracts features from each detected object using a convolutional neural network. These features represent the object's appearance and are robust to variations in perspective, lighting, and partial occlusions. After feature extraction, the algorithm performs a matching step. Here, a Kalman filter is employed for predicting the new locations of tracked objects in the current frame. These predictions are then matched with the new detections based on the calculated similarity scores derived from the extracted features. The similarity scores ensure that the tracked objects are consistently identified across successive frames, thereby maintaining their identities. We present the DeepSport in Algorithm 1.

In cases where objects leave the field of view and reappear, DeepSort's feature memory aids in re-identifying these objects. This capability is crucial for long-term tracking in complex environments where objects frequently interact or get occluded. Additionally, DeepSort incorporates a mechanism to handle the birth and death of tracks, efficiently managing the initiation of new object tracks and the termination of those no longer visible. This aspect of the algorithm ensures efficient use of computational resources and minimizes the chances of false track continuations.

The process of Algorithm 1 is as follows.

1. **Initialization:** An empty list for tracks (`track_list`) is initialized. This list will store the tracking information for each detected object.

2. **Frame Iteration:** The algorithm iterates over each frame in the set of video frames.

3. **Object Detection:** For each frame, object detections are obtained using a YOLO model. These detections include bounding boxes, class labels, and confidence scores.

4. **Detection Iteration:** The algorithm processes each detection found in the current frame.

5. **Detection Details Extraction:** For each detection, the bounding box, class label, and confidence score are extracted.

6. **Feature Extraction:** Features are extracted for each detection's bounding box, crucial for tracking the object in subsequent frames.

7. **Track Prediction:** The algorithm iterates over each track in `track_list` and predicts the current position of each track using a Kalman Filter.

8. **Data Association:** Detections are associated with existing tracks. The outcome includes matched pairs of tracks and detections, unmatched tracks, and unmatched detections.

9. **Track Update:** For each matched pair, the corresponding track is updated with the new detection information.

10. **New Track Creation:** New tracks are created for detections that did not match any existing track.

Algorithm 1: Pseudo-Code for DeepSort Algorithm

```
Result: Track objects in video frames
initialization: Initialize an empty list for tracks:
  track_list;
for each frame in video_frames do
    detections ←
      get_detections_from_YOLO(frame);
    for each detection in detections do
        Extract bbox, class_label, confidence;
        feature ← extract_features(bbox, frame);
    end
    for each track in track_list do
        track.predict();
    end
    matches, unmatched_tracks,
      unmatched_detections ←
      associate_data(track_list, detections);
    for each match in matches do
        track, detection ← match;
        track.update(detection);
    end
    for each detection in unmatched_detections do
        new_track ←
          create_new_track(detection);
        track_list.append(new_track);
    end
    for each track in unmatched_tracks do
        if track.is_confirmed() and
          track.time_since_update > max_age then
            track_list.remove(track);
        end
    end
    output_tracks ← [track for track in track_list
      if track.is_confirmed()];
    visualize_tracking_results(frame,
      output_tracks);
end
```

11. **Adding New Tracks:** Each new track is added to the `track_list`.

12. **Unmatched Track Handling:** The algorithm checks each unmatched track to determine if it should be removed based on a specified condition, like not being updated for a certain time (`max_age`).

13. **Confirmed Tracks Output:** The algorithm compiles a list of confirmed tracks from `track_list`.

14. **Visualization:** The tracking results, including the paths and identities of the tracked objects, are visualized on the current frame.

3.3. Combination of Yolov8 and DeepSort

Integrating YOLOv8 with DeepSort creates a powerful real-time object detection and tracking system, particularly useful in surveillance, autonomous driving, and activity monitoring scenarios. The integration of these two algorithms hinges on the seamless flow of information from the object detection phase (handled by YOLOv8) to the tracking phase (managed by DeepSort) [23, 24].

The process begins with YOLOv8, which takes as input video frames. YOLOv8, known for its speed and accuracy in object detection, processes these frames to identify and localize objects, see Algorithm 1. It analyzes the visual data and outputs a set of bounding boxes for each detected object. Each bounding box is accompanied by a confidence score indicating the likelihood of the object's presence and its class label (like a person, car, etc.). This stage is critical because the quality of object detection directly impacts the subsequent tracking performance. Once YOLOv8 detects objects in a frame, the information is passed to DeepSort. DeepSort takes the bounding boxes generated by YOLOv8 as its input. The first step in DeepSort is feature extraction. DeepSort extracts a unique feature vector for each detected object using a CNN. This feature vector essentially captures the object's appearance and is used to distinguish it from others. Following feature extraction, DeepSort employs a Kalman Filter [27] to predict the trajectory of each tracked object. The Kalman Filter estimates the current position and velocity of the objects based on the observed states in previous frames. This predictive step is crucial, especially in cases of temporary occlusion or erratic object movement.

The next phase is data association, where DeepSort matches the current frame's detected objects with existing tracks. This matching is based on the spatial proximity (the location of the bounding boxes) and the similarity of the feature vectors (appearance). The Hungarian algorithm, an optimization technique, is typically used for this assignment process, ensuring that each detected object is matched to the correct track. Lastly, DeepSort updates the tracks with the matched detections and manages them by creating new tracks for unmatched detections and terminating old tracks that have disappeared or have been inactive for a certain period. The output of this integrated system is a continuous and coherent track for each detected object throughout the video, even in the presence of challenges like occlusion, interaction, or varied object motion.

4. Experiments

4.1. Yolov8 models

In these experiments, the researchers initially acquire pre-trained detection models, along with their weights, to serve as the basis for further training on a custom dataset. These models were previously trained on the COCO dataset, which includes 80 classes. Table 1 provides a comparative overview of different YOLOv8 models in terms of their complexity and computational requirements.

Table 1. Brief information on pre-trained YOLOv8 models.

Model	# layers	# params	# gradients	# GFLOPS
YOLOv8n	225	3157200	3157184	8.9
YOLOv8s	225	11135987	11135971	28.6
YOLOv8m	295	25856899	25856883	79.1
YOLOv8l	365	43630611	43630595	165.4
YOLOv8x	365	68153571	68153555	258.1

4.2. Datasets

The research team compiled 100 screenshots from each fish farming and chicken farming environment. Since the YOLOv8 models are trained using their original weights instead of starting from scratch, there's no need for a large dataset. This approach leverages an important advantage known as transfer learning [28, 29], which allows for the training of models on custom datasets to refine pre-existing weights [30]. Subsequently, labels were created for these datasets. The data was split into training, validation, and testing sets with proportions of 80%, 10%, and 10%, respectively. These datasets have been made publicly accessible[2][3] to foster further reproducibility and utilization.

4.3. Loss Function Optimization

The loss function for a YOLO model that incorporates box_loss, cls_loss, and dfl_loss can be represented as a combination of these individual loss components.

$$\text{total\_loss} = \lambda_{\text{box}} \cdot \text{box\_loss} + \lambda_{\text{cls}} \cdot \text{cls\_loss} + \lambda_{\text{dfl}} \cdot \text{dfl\_loss} \tag{1}$$

where:

$$\begin{aligned}\text{box\_loss} = \sum_{i=0}^{S^2} \sum_{j=0}^{B} 1_{ij}^{\text{obj}} \Big[&(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2 \\ &+ (w_i - \hat{w}_i)^2 + (h_i - \hat{h}_i)^2 \Big]\end{aligned} \tag{2}$$

[2] https://app.roboflow.com/dfki/chicken-6h00m/overview
[3] https://app.roboflow.com/dfki/fish-e375s/overview

$$\text{cls\_loss} = \sum_{i=0}^{S^2} 1_i^{\text{obj}} \sum_{c\in\text{classes}} \left(p_i(c) - \hat{p}_i(c)\right)^2 \quad (3)$$

$$\text{dfl\_loss} = \sum_{i=0}^{S^2} 1_i^{\text{obj}} \cdot \text{FocalLoss}(f_i, \hat{f}_i) \quad (4)$$

Total Loss total_loss: The overall loss is a weighted sum of the box loss, classification loss, and DFL loss. The weights λ_{box}, λ_{cls}, and λ_{dfl} are hyperparameters that balance the contribution of each component.

Box Loss box_loss: This part calculates the mean squared error between the predicted and actual bounding box coordinates (x, y, w, h). The sum runs over all grid cells (S^2) and bounding boxes per cell (B). The indicator function 1_{ij}^{obj} is 1 if an object is present in the jth bounding box of cell i, otherwise 0.

Classification Loss cls_loss: This is typically a sum of squared errors (or could be cross-entropy) for the class predictions over all classes. The indicator 1_i^{obj} is 1 if an object is present in cell i.

DFL Loss dfl_loss: This represents the Distance Focal Loss. The exact form of this loss depends on its specific definition, which might vary. Here, it's represented as a generic focal loss function applied to some feature f, with $\hat{f}_i$ representing the predicted feature. Note that focal loss addresses class imbalance issues in object detection (where some classes are much more frequent than others). Distance Focal Loss would be a variant of this, possibly focusing on the distance aspect of the bounding box predictions or object features. It's meant to give more weight to harder, misclassified examples so the model learns to handle them better.

4.4. Evaluation Metrics

The metrics mAP50 and mAP50-95 are commonly used to report the performance of object detection models. These metrics are variations of the mean Average Precision (mAP).

The mean Average Precision at an IOU of 50% (mAP50) is calculated as follows:

$$\text{mAP50} = \frac{1}{N} \sum_{i=1}^{N} \text{AP}_{50}(C_i) \quad (5)$$

where:

- N is the number of classes.
- C_i is the i-th class.
- $\text{AP}_{50}(C_i)$ is the Average Precision for class C_i at an IOU threshold of 50%.

IOU stands for Intersection Over Union, a measure of overlap between the predicted bounding box and the ground truth box. An IOU of 50% means that the predicted box should overlap at least 50% with the ground truth for a detection to be considered correct.

The mAP50-95 is the mean of the AP calculated at different IOU thresholds ranging from 50% to 95%, in steps of 5%:

$$\text{mAP50-95} = \frac{1}{N} \sum_{i=1}^{N} \frac{1}{T} \sum_{t=50\%}^{95\%} \text{AP}_t(C_i) \quad (6)$$

where:

- N is the number of classes.
- C_i is the i-th class.
- T is the number of IOU thresholds (from 50% to 95%, in 5% increments).
- $\text{AP}_t(C_i)$ is the Average Precision for class C_i at an IOU threshold of t.

4.5. Hyperparameters

In our experiments, we explored various hyperparameter combinations and discovered that the default settings[4] for YOLOv8 and Deepsort were not optimal for achieving the highest scores. Consequently, we made the following adjustments:

1. **Optimizer:** Switched from SGD to AdamW.
2. **Dropout Rate:** Increased from 0.0 to 0.2.
3. **Initial Learning Rate:** Decreased from 0.01 to 0.001.
4. **Final OneCycleLR Learning Rate:** Adjusted from 0.01 to 0.1.
5. **Image Translation:** Modified from 0.1 to 0.0.
6. **Image Flip (Left-Right Probability):** Altered from 0.5 to 0.0.
7. **Image Scale:** Changed from 0.5 to 0.0.

The experiments were carried out on a workstation featuring a 13$^{\text{th}}$ Generation Intel® Core™ i5-13600KF CPU, which has 20 logical processors. The data loader utilized all the available cores in this setup. The system was also equipped with 32GB of RAM and an NVIDIA GeForce RTX 3060 graphics card with 12GB of memory.

[4] https://github.com/ultralytics/ultralytics/blob/main/ultralytics/cfg/default.yaml

4.6. Experimental Results

The provided Table 2 showcases the performance of our YOLOv8-agri models that have been fine-tuned on custom agricultural datasets targeting chicken and fish. The models are specialized adaptations of the original YOLOv8, optimized for domain-specific data.

In the chicken dataset, the YOLOv8l-agri model stands out by achieving the highest mAP50 score at **0.871**, indicating it has the best performance for object detection with at least 50% IOU. This model also excels in the more comprehensive mAP50-95 metric with a leading score of **0.491**, showcasing its consistent accuracy across a range of IOU thresholds. However, this high level of precision comes with a relatively longer training time of 0.822 hours. On the other end, the YOLOv8x-agri model, while demonstrating a commendable mAP50 score, requires the most extended training time in the group, clocking in at 0.994 hours. This suggests a potential trade-off between the model's training efficiency and its detection performance.

For the fish dataset, the YOLOv8l-agri model confirms its superior detection capabilities with the highest mAP50 score of **0.718**, demonstrating robust performance across different domains. Contrasting with its performance on the chicken dataset, the YOLOv8x-agri model achieves the best mAP50-95 score of **0.359** for the fish data. This indicates that while it may not always secure the top mAP50 score, it performs better when considering the entire spectrum of IOU thresholds. Notably, the training times for models on the fish dataset are longer than those on the chicken dataset, which could reflect the inherent complexity or the challenges presented by the fish dataset. The YOLOv8x-agri model exhibits the longest training duration of 0.895 hours among the models evaluated for the fish data.

In conclusion, the YOLOv8l-agri model is distinguished as the most effective for the given agricultural and fishery contexts, balancing detection precision and training time efficiency. While the YOLOv8x-agri does not consistently achieve the highest mAP50 scores, its performance in the mAP50-95 metric, especially for the fish dataset, is noteworthy and may be crucial for applications where precise object localization is critical. These findings emphasize the significance of tailoring model selection to specific domain needs and highlight the applicability of the YOLOv8-agri models in the field of precision agriculture, where accuracy is of utmost importance despite the potential for increased training times. Additionally, we present several screenshots of the test videos in Figures 4 and 5.

Table 2. The performance of our YOLOv8-agri models on chicken and fish datasets. The best scores are in bold.

Model	chicken data			fish data		
	mAP50	mAP50-95	training time (h)	mAP50	mAP50-95	training time (h)
YOLOv8n-agri	0.790	0.444	0.734	0.656	0.326	0.729
YOLOv8s-agri	0.852	0.463	0.749	0.668	0.322	0.733
YOLOv8m-agri	0.813	0.470	0.799	0.696	0.358	0.816
YOLOv8l-agri	**0.871**	**0.491**	0.822	**0.718**	0.334	0.826
YOLOv8x-agri	0.792	0.441	0.994	0.717	**0.359**	0.895

Figure 4. The detection performance of our Yolov8-agri models on the screenshot of a fish farming. The number presents the unique object ID tracked by DeepSort.

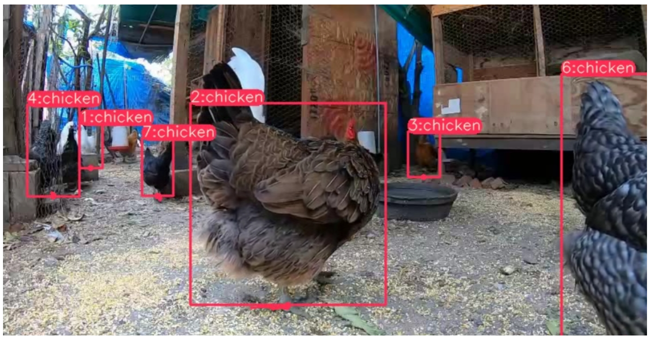

Figure 5. The detection performance of our Yolov8-agri models on the screenshot of a chicken farming. The number presents the unique object ID tracked by DeepSort.

To promote future reproducibility, we have made our YOLOv8-agri models publicly available, complementing existing public datasets. These can be accessed online at `https://github.com/duongtrung/YOLOv8-agri`.

5. Remarks and Discussion

In this section, we delve into the nuanced application of YOLOv8 in the agriculture and fisheries sectors, where its advanced object detection capabilities are tested against the unique challenges these settings pose. Despite YOLOv8's prowess in identifying a wide array of objects, it needs help distinguishing between closely related categories, such as differentiating between various types of fish and chickens, often misidentifying them as generic birds. This limitation underscores the pressing need for specialized training datasets to encapsulate the distinctive features of targeted categories, ensuring the model can accurately classify the unique attributes in these industries.

While YOLOv8 excels in general computer vision tasks, deploying it in agriculture and fisheries requires significant customization. The model must be trained with datasets rich in domain-specific details to improve its accuracy in identifying and classifying the complex variety of objects encountered in these fields. This approach will enable YOLOv8 to overcome its inherent limitations, offering more precise and reliable object detection essential for the nuanced demands of agricultural and fisheries operations.

Furthermore, our analysis reveals a conspicuous research gap in integrating YOLOv8 with motion-tracking algorithms like DeepSORT, especially in agriculture. Prior research has yet to fully explore the potential of leveraging YOLOv8's detection accuracy with DeepSORT's tracking capabilities to enhance precision monitoring in these sectors. Our proposed solution aims to fill this gap by combining the strengths of YOLOv8 and DeepSORT. This integration provides enhanced object recognition and tracking performance, explicitly tailored to the peculiarities of the agriculture and fisheries industries, thereby enabling more effective and efficient monitoring and management practices. By combining YOLOv8's detection capabilities with DeepSORT's tracking efficiency, we propose a novel approach that could significantly impact precision agriculture and fisheries monitoring. This innovative solution promises to improve the accuracy of object detection and tracking and facilitate real-time monitoring and management of resources, contributing to the sustainability and productivity of these crucial sectors. Through this pioneering integration, we aim to address the specific challenges faced in agriculture and fisheries, demonstrating the transformative potential of advanced computer vision and tracking technologies.

Our methodological approach also introduces an efficient data utilization strategy. By compiling 100 screenshots each from fish and chicken farming scenarios and utilizing transfer learning, we capitalize on YOLOv8 models pre-trained on extensive datasets. This underscores the advantage of the fine-tuning approach, eliminating the necessity for large-scale data acquisition. Rather than starting from the beginning, the models are adapted to identify new labels. This method is especially beneficial in the agricultural sector, as it reduces the requirement for farmers or developers to invest in advanced computing equipment. The process of retraining with new datasets is swift, often taking just a few hours. Moreover, the authors have provided their experimental datasets for additional comparison, positioning our set of 100 images as a robust baseline in its present state.

In our comparative analysis of the YOLOv8-agri models, the YOLOv8l-agri emerges as a balanced option, optimizing both precision and training efficiency within our targeted context. Although YOLOv8x-agri may not consistently outperform in mAP50 scores, its proficiency in the mAP50-95 metric, particularly for fish data, is compelling. Such precision is indispensable in scenarios where exact object localization is critical, despite potentially longer training durations. Our findings advocate for a strategic selection of models based on domain-specific requisites, reinforcing the viability of YOLOv8-agri models for precision agriculture where accuracy is paramount.

6. Conclusion

In conclusion, the fusion of YOLOv8-agri and DeepSORT algorithms embodies a groundbreaking step towards enhancing computer vision applications in agriculture and fisheries. Our investigations confirm that the YOLOv8l-agri model excels in achieving an equilibrium between accuracy and training time efficiency, proving to be invaluable for real-time precision tasks. Importantly, we have contributed to the scientific community by providing open access to our experimental datasets and trained models, thus laying the groundwork for reproducibility and future research endeavors. This transparency is essential for the continuous improvement and validation of technology in specialized sectors, ultimately contributing to smarter, more sustainable agricultural practices and fishery management.

References

[1] Bi, Y., Xue, B., Mesejo, P., Cagnoni, S. and Zhang, M. (2022) A survey on evolutionary computation for computer vision and image analysis: Past, present, and future trends. *IEEE Transactions on Evolutionary Computation* **27**(1): 5–25.

[2] Mahadevkar, S.V., Khemani, B., Patil, S., Kotecha, K., Vora, D., Abraham, A. and Gabralla, L.A. (2022) A review on machine learning styles in computer vision-techniques and future directions. *IEEE Access* .

[3] Dhanya, V., Subeesh, A., Kushwaha, N., Vishwakarma, D.K., Kumar, T.N., Ritika, G. and Singh, A. (2022) Deep learning based computer vision approaches for smart agricultural applications. *Artificial Intelligence in Agriculture* .

[4] Punithavathi, R., Rani, A.D.C., Sughashini, K., Kurangi, C., Nirmala, M., Ahmed, H.F.T. and Balamurugan, S. (2023) Computer vision and deep learning-enabled weed detection model for precision agriculture. *Comput. Syst. Sci. Eng* **44**(3): 2759–2774.

[5] Huynh, H.X., Tran, L.N. and Duong-Trung, N. (2023) Smart greenhouse construction and irrigation control system for optimal brassica juncea development. *Plos one* **18**(10): e0292971.

[6] Hu, W.C., Chen, L.B., Huang, B.K. and Lin, H.M. (2022) A computer vision-based intelligent fish feeding system

using deep learning techniques for aquaculture. *IEEE Sensors Journal* 22(7): 7185–7194.

[7] Saleh, A., Sheaves, M. and Rahimi Azghadi, M. (2022) Computer vision and deep learning for fish classification in underwater habitats: A survey. *Fish and Fisheries* **23**(4): 977–999.

[8] Gladju, J., Kamalam, B.S. and Kanagaraj, A. (2022) Applications of data mining and machine learning framework in aquaculture and fisheries: A review. *Smart Agricultural Technology* **2**: 100061.

[9] Jocher, G., Chaurasia, A. and Qiu, J. (2023), Ultralytics yolov8. URL https://github.com/ultralytics/ultralytics.

[10] Yamada, Y. and Otani, M. (2022) Does robustness on imagenet transfer to downstream tasks? In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*: 9215–9224.

[11] Luccioni, A.S. and Rolnick, D. (2023) Bugs in the data: How imagenet misrepresents biodiversity. In *Proceedings of the AAAI Conference on Artificial Intelligence*, **37**: 14382–14390.

[12] Terven, J. and Cordova-Esparza, D. (2023) A comprehensive review of yolo: From yolov1 to yolov8 and beyond. *arXiv preprint arXiv:2304.00501* .

[13] Diwan, T., Anirudh, G. and Tembhurne, J.V. (2023) Object detection using yolo: Challenges, architectural successors, datasets and applications. *multimedia Tools and Applications* **82**(6): 9243–9275.

[14] Mei, Y., Sun, B., Li, D., Yu, H., Qin, H., Liu, H., Yan, N. *et al.* (2022) Recent advances of target tracking applications in aquaculture with emphasis on fish. *Computers and Electronics in Agriculture* **201**: 107335.

[15] Coulibaly, S., Kamsu-Foguem, B., Kamissoko, D. and Traore, D. (2022) Deep learning for precision agriculture: A bibliometric analysis. *Intelligent Systems with Applications* **16**: 200102.

[16] Paul, A., Ghosh, S., Das, A.K., Goswami, S., Das Choudhury, S. and Sen, S. (2020) A review on agricultural advancement based on computer vision and machine learning. *Emerging Technology in Modelling and Graphics: Proceedings of IEM Graph 2018* : 567–581.

[17] Qin, Z., Wang, W., Dammer, K.H., Guo, L. and Cao, Z. (2021) Ag-yolo: A real-time low-cost detector for precise spraying with case study of palms. *Frontiers in Plant Science* **12**: 753603.

[18] Kandimalla, V., Richard, M., Smith, F., Quirion, J., Torgo, L. and Whidden, C. (2022) Automated detection, classification and counting of fish in fish passages with deep learning. *Frontiers in Marine Science* **8**: 2049.

[19] Durve, M., Orsini, S., Tiribocchi, A., Montessori, A., Tucny, J.M., Lauricella, M., Camposeo, A. *et al.* (2023) Benchmarking yolov5 and yolov7 models with deepsort for droplet tracking applications. *The European Physical Journal E* **46**(5): 32.

[20] Paik, C. and Kim, H.J. (2022) Improving object detection, multi-object tracking, and re-identification for disaster response drones. *arXiv preprint arXiv:2201.01494* .

[21] Razzok, M., Badri, A., El Mourabit, I., Ruichek, Y. and Sahel, A. (2023) Pedestrian detection and tracking system based on deep-sort, yolov5, and new data association metrics. *Information* **14**(4): 218.

[22] Wang, Y. and Yang, H. (2022) Multi-target pedestrian tracking based on yolov5 and deepsort. In *2022 IEEE Asia-Pacific Conference on Image Processing, Electronics and Computers (IPEC)* (IEEE): 508–514.

[23] Vats, A. and Anastasiu, D.C. (2023) Enhancing retail checkout through video inpainting, yolov8 detection, and deepsort tracking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*: 5529–5536.

[24] Sharma, N., Baral, S., Paing, M.P. and Chawuthai, R. (2023) Parking time violation tracking using yolov8 and tracking algorithms. *Sensors* **23**(13): 5843.

[25] Wojke, N., Bewley, A. and Paulus, D. (2017) Simple online and realtime tracking with a deep association metric. In *2017 IEEE international conference on image processing (ICIP)* (IEEE): 3645–3649.

[26] Bewley, A., Ge, Z., Ott, L., Ramos, F. and Upcroft, B. (2016) Simple online and realtime tracking. In *2016 IEEE international conference on image processing (ICIP)* (IEEE): 3464–3468.

[27] Khodarahmi, M. and Maihami, V. (2023) A review on kalman filter models. *Archives of Computational Methods in Engineering* **30**(1): 727–747.

[28] Duong-Trung, N., Quach, L.D. and Nguyen, C.N. (2019) Learning deep transferability for several agricultural classification problems. *International Journal of Advanced Computer Science and Applications* **10**(1).

[29] Duong-Trung, N., Le Ha, D.N. and Huynh, H.X. (2021) Classification-segmentation pipeline for mri via transfer learning and residual networks. In *RICE*: 39–43.

[30] Duong-Trung, N., Quach, L.D., Nguyen, M.H. and Nguyen, C.N. (2019) Classification of grain discoloration via transfer learning and convolutional neural networks. In *Proceedings of the 3rd International Conference on Machine Learning and Soft Computing*: 27–32.