From YOLO V1 to YOLO V11: comparative analysis of YOLO algorithm (review)

Imane Beqqali Hassani[1], Soufia Benhida[1], Nabil Lamii[1], Khalid Oqaidi[1], Ahmed Ouiddad[2], Soukaina Ghiadi[3]

[1]Research, Development, and Innovation Laboratory, Mundiapolis University, Casablanca, Morocco
[2]LAMSAD Laboratory, National School of Applied Sciences, Hassan 1st University, Berrechid, Morocco
[3]Interdisciplinary Laboratory of Applied Sciences, National School of Applied Sciences, Hassan 1st University, Berrechid, Morocco

Article Info

Article history:

Received Apr 1, 2025
Revised Oct 24, 2025
Accepted Nov 23, 2025

Keywords:

Computer vision
Evolution of YOLO
Machine learning
Object-based detection
YOLO algorithm

ABSTRACT

Object detection in images or videos faces several challenges because the detection must be accurate, efficient and fast. The you only look once (YOLO) algorithm was invented to meet these criteria. But with the creation of several versions of this algorithm (from V1 to V11), it becomes difficult for researchers to choose the best one. The main objective of this review is to present and compare the eleven versions of the yolo algorithm in order to know when using the appropriate one for the study. The methodology used for this work is aligned with preferred reporting items for systematic reviews and meta-analyses (PRISMA) principles and the results demonstrate that the choice of the best version mainly depends on the priorities of the study. If the study prioritizes accuracy and detection of small objects, it should use YOLO V4, YOLO V5, YOLO V6, YOLO V7, YOLO V8, YOLO V9, YOLO V10 or YOLO V11. While studies that prioritize detection speed should use YOLO V5, YOLO V6, YOLO V7, YOLO V8, YOLO V10 or YOLO V11. In complex environment, researchers should avoid using YOLO V1, YOLO V2, YOLO V3, YOLO V5, YOLO V7 and YOLO V9. And researchers who are looking for a good accuracy and speed and a reduced number of parameters should use YOLO V10 or YOLO V11.

Corresponding Author:

Imane Beqqali Hassani
Research, Development, and Innovation Laboratory, Mundiapolis University
Casablanca, Morocco
Email: i.beqqali@gmail.com

1. INTRODUCTION

Deep learning has advanced significantly in recent years and has contributed to the advancement of several fields (health, education, biology, aeronautics, and automotive). This is due to the availability of resources such as datasets, robustness of hardware and the development of software and work tools. However, there is still a lot of space for research and development [1], especially in the field of real-time object detection. Real-time is critical, because it does not tolerate any margin of error. Objects must be detected one hundred percent if we want to have a perfect, consistent and reliable result. To meet these requirements, researchers have used the YOLO algorithm because according to Alahdal *et al.* [2], different versions of YOLO have demonstrated speed and accuracy in detecting objects. Since 2016, the YOLO algorithm has evolved and researchers have been able to create eleven versions of this algorithm (from YOLO V1 to YOLO V11). Each of them has its own characteristics, advantages and disadvantages.

Several studies in the domain of computer vision and real-time detection emphasize the importance of striking an equilibrium between precision and speed. However, previous work tends to focus only on a single version of the YOLO algorithm and fails to offer comparative studies of multiple YOLO versions (especially those including recent versions) in order to provide concrete recommendations. By covering all versions of YOLO up to version 11, this literature review fills a critical gap in the literature and addresses the practical needs of researchers in selecting the best object detection model based on their requirements.

In this context, we present this review in order to answer to the following research question: “What are the changes made to the YOLO algorithm in each of its versions, what are the advantages and disadvantages of every YOLO algorithm version and what is the best version to use for every requirement?”. To answer this question, we propose the present work, whose objective is to provide a comparative analysis of the different versions of the YOLO algorithm. More precisely, we identify the advantages and disadvantages and present the contribution of each version of YOLO in terms of architecture, accuracy, speed, and improvements in order to know the best version to use for every requirement. Thus, the present paper is structured as follows: as a first step, we describe the methodology of this literature review. Then we describe an overview about YOLO algorithm by presenting its eleven versions. After that, we report the results and finally we discuss them.

2. METHODOLOGY

The methodology of this review is based on the methodology of [3] aligned with PRISMA principles [4]. It includes the following steps: i) Research questions for the literature review, ii) Document search strategy, iii) Document selection criteria, iv) Document validity assessment and v) Data collection. In this review we answer the following research questions as shown in Table 1 whose formulation was made based on the population, intervention, comparison, outcome (PICO) technique [5]:

Table 1. Research questions based on the PICO technique

Research question	Corresponding PICOC element
What are the changes made to the YOLO algorithm in each of its versions?	Population: YOLO Algorithm versions Intervention: Changes and modifications across versions
Which version of the yolo algorithm is the best?	Comparison: Comparison between versions to determine the best one
What are the advantages and disadvantages of each version of the YOLO algorithm?	Outcome: Advantages/Disadvantages

2.1. Search strategy and selection process

The articles used in this review are English written, extracted from ScienceDirect and google scholar databases and were published between 2012 and 2025 as shown in Figure 1. The majority of the articles are very recent. Indeed 3 articles were published in 2025, 29 articles were published in 2024, 5 articles were published in 2023, 3 articles were published for each year 2018, 2020, 2021 and 2022. And for 2012, 2013, 2014, 2015, 2016 and 2017 only one article is used for each year. The articles use the YOLO algorithm applied in different fields and in different countries.

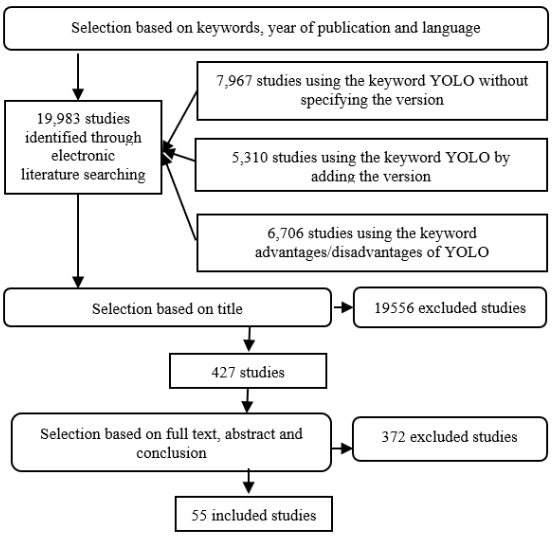

Figure 1. Research procedure

The keywords/search strings use advanced Boolean queries on scientific databases and are as: i) YOLO algorithm OR Yolo OR Object detection using YOLO algorithm. ii) YOLO version (Number) OR YOLO V (Number) OR YOLO V (Number). And iii) Advantages OR/AND disadvantages of YOLO V (Number). We replace the (Number), with a number from 1 to 11 for each search.

The following criteria serve as the foundation for the selection process:

- Year of publication: the most recent articles are the most preferred
- English language: Only English-language publications are used.
- Title: only articles mentioning "YOLO", "Object detection" in the title are reviewed.
- Finally, we exclude studies that do not address object detection, and that do not use the YOLO algorithm.

2.2. Document validity assessment and data collection

In order to evaluate the quality of studies, we use questions, and we assign a score to each question. Accordingly, the values "0," "0.5," and "1" are assigned to the answers "No," "Partially," and "Yes," respectively. The sum of the scores for each study is then calculated. The quality assessment questions (Q) are as follows: (Q1) Does the study provide an explicit use of the YOLO algorithm? (Q2) Is object detection the main objective of the study?

For articles that discussed the YOLO algorithm, we extracted the following data: year of publication, version of the YOLO algorithm, architecture, study results, and advantages and disadvantages of the used version.

3. YOLO ALGORITHM: AN OVERVIEW

According to Gheorghe *et al.* [6], YOLO algorithm has becoming very popular among the series of object detection models. The algorithm has been improved significantly since its first publication, with versions spanning from V1 to V11 [7]. During its evolution, each version of YOLO has been improved to have more precision [8], smaller volume and higher speed [7].

The first two versions of yolo use general architectures, but starting with YOLO V3, the main architecture of the YOLO algorithm includes three main parts, namely the backbone, neck and head [9]. The main role of the backbone is to extract the most important features of the image and transmit them to the head through the neck [9]. The main function of the neck is very important in this process. Because it compiles the feature maps generated by the backbone [9]. In addition to this, the neck helps improving received image's features, by building feature pyramids that help making multiscale object detection easier [9]. Once processing is completed, the neck sends the data to the head, which in turn makes the final predictions in terms of classification and generalization.

3.1. YOLO V1

First introduced by Redmon *et al.* [10], YOLO V1 is based on convolutional neural networks (CNNs) to detect object through using regression [11]. It directly improves detection performance by training on complete images [10]. The one-state object detection mode is mainly used in YOLO V1. This allows to simultaneously predict all bounding boxes with their classes using just a single pass through the model [11].

The network architecture of YOLO V1 is inspired by the GoogLeNet model [12] for image classification [10]. This first version of YOLO uses a grid for object detection. Each cell of the grid allows the prediction of the bounding boxes and classes, knowing that each entry is divided into S x S [8]. The architecture of YOLO V1 consists of 24 convolutional layers with 2 fully connected layers [8]. To reduce the number of required channels, each layer uses a 1 x 1 convolution that precedes a 3 x 3 convolution. Except the last layer that uses a linear activation function, each layer of YOLO V1 uses leaky rectified linear unit (LeakyReLU) as its activation function [8].

However, the performance of YOLO V1 is sub-optimal in scenarios where objects are close to each other [9]. The new versions of YOLO are more efficient and faster but the first version of this algorithm represents a real leap forward in the history of object detection. Indeed, according to [9], in the history of computer vision, YOLO V1 is a significant accomplishment since it opened the door for other sophisticated detection algorithms.

3.2. YOLO V2

Released by Redmon and Farhadi in 2017, YOLO V2 is building upon YOLO V1, and often referred to as YOLO9000. The researchers called it yolo9000 because it can detect over 9,000 object categories [13]. It optimizes detection and classification [13] and focuses primarily on improving object localization and recall to enhance object detection performance [11]. Using a single neural network, the second version of YOLO directly predicts the bounding box and the associated category [14]. The model

YOLO V2 utilizes anchor boxes to predict bounding boxes [13]. YOLO V2's anchor ideas build on that of faster region-based convolutional neural network (Faster R-CNN), which samples on the convolution feature map with sliding window [15]. According to Yan *et al.* [11], in the training set, clustering the bounding boxes into instances allows determining the sizes of the anchor boxes by using K-means.

The structure of YOLO V2 is based on the darknet framework which is similar to the architecture of YOLO V1 [8]. The difference between the structure of YOLO V1 and that of YOLO V2 is that for YOLO V2 we find only 19 convolutional layers which precede 5 max pooling layers and the activation function rectified linear unit (ReLU) is used for each convolutional layer [8]. In addition to that, YOLO V2 uses batch normalization after each convolutional layer, which helps to accelerate convergence, increase accuracy and improve training stability.

3.3. YOLO V3

Created by Farhadi and Redmon in 2018, YOLO V3 featured significantly more advancements. Indeed, compared to YOLO V2, the Average accuracy is essentially improved in YOLO V3 [16]. According to Yan *et al.* [11], for each bounding box, in order to estimate the objectness score, YOLO V3 uses logistic regression. The structure of YOLO V3 is mostly made up of the feature pyramid network (FPN) and the darknet-53 backbone [17]. Indeed, the backbone network architecture used in YOLO V3 is the Darknet-53 architecture [18] which consists of 53 convolutional layers, knowing that each layer uses a leakyReLU activation function [8]. In addition, to form anchor boxes, K-means clustering is used by YOLO V3 [19] and has the ability to predict bounding boxes on three different scales [11]. YOLO V3 differs from other object detection algorithms that allow region generation because it returns the direct position of the bounding box and its category by using the entire image as the network's input [19]. YOLO V3 incorporates the latest technologies such as up-sampling, skip-connection, and residual blocks. DarkNet-53 is able to obtain more scaling feature maps, such as the small, medium, and large scales of target features [19]. Multi-scale predictions and residual network structure can perform better feature extraction, and improve the mean average precision (mAP) and small object detection results [19]. In order to detect multiple objects with different classes in the same grid and train logistic classifiers for classification, YOLO V3 uses binary cross-entropy. That enables to label objects that are in the same grid [8].

3.4. YOLO V4

YOLO V4 is the first version of YOLO created in 2020 by Bochkovskiy *et al.* and in which Redmon was not involved. To achieve its objective, the network architecture of YOLO V4 uses several components ([11], [20]). Like YOLO V3, the architecture of YOLO V4 consists of 53 convolutional layers. The difference between both architectures is that YOLO V4 uses for each layer the mish activation function and allows the usage of cross-stage partial connections [8]. The backbone includes the architecture of cross-stage partial CSPDarknet-53 [21]. Compared to the YOLO V3 approach, the learning capability of the CNN is improved througth CSPDarknet53 backbone model [8]. According to Fahim and Hasan [9], in CSPDarkNet53, the feature map of the base layer is divided into 2 sections and then merged by using a cross stage partial network (CSPNet). The neck includes the path-aggregation networks (PANet) [22] and the additional spatial pyramid pooling (SPP) module [23]. In order to improve accuracy and reduce computational overhead, the traditional feature pyramid network (FPN) neck used in YOLO V3 has been replaced in YOLO V4 by the PANet neck for parameter aggregation [8]. The head includes the YOLO V3 anchor-based architecture.

According to Xu *et al.* [24], using mosaic data augmentation improved learning much better in YOLO V4. This technique allows combining 4 original input images in order to create a new training image so that the network learn recognizing some objects that are not found in their usual environment [8]. Each image takes up a quarter of the final image, and the objects (bounding boxes) are readjusted accordingly in order to increase the variety of data and improve the model's ability to generalize.

Providing an ideal equilibrium between object detection accuracy and computational complexity is a significant asset for YOLO V4 and that represents an incremental improvement over YOLO V3 [8]. The implementation of diverse training techniques that are part of the YOLO V4 network architecture allows obtaining an improved performance and faster processing [11]. In addition to that, YOLO V4 is designed to be trainable on a single graphics processing unit (GPU) card, making it accessible to researchers with limited resources.

3.5. YOLO V5

Glenn Jocher introduced the YOLO V5 in June 2020 [25], which has some notable improvements and some distinctions [9]. The key innovation of YOLO V5 is that it is the 1st version of YOLO built using PyTorch in python and not based on darknet architecture, allowing for a more straightforward implementation process and development [8]. YOLO V5 integrated the anchor box selection process [26].

Indeed, for objects in the dataset, the distribution of the bounding box location contributes to auto learning anchor boxes [8].

Several network architectures are proposed by YOLO V5 and are suitable for several scenarios with various input sizes such as YOLO V5 s, YOLO V5 m and YOLO V5 l [11]. In each of these models the structure of YOLO V5 differs greatly in terms of width and depth of each convolutional module [8]. According to Ahmed *et al.* [7], these variants of YOLO V5 offer varying performance and detection accuracy, achieved by adjusting layer count and network depth.

There is a great similarity between the architecture of YOLO V5 and that of YOLO V4 [11]. The three main components of the architecture of YOLO V5 are: the head for predicting classes and bounding boxes, the PANet neck for collecting feature maps, and the cross stage partial (CSP) backbone for extracting image features ([8], [13], [27]). YOLO V5 uses a modified cross stage partial darknet-53(CSPDarkNet53) backbone [8]. In order to detect objects of various sizes, CSPDarkNet53 proposes 53 convolutional layers and generates features at multiple scales with a modified PANet [8].

3.6. YOLO V6

In 2022, Li *et al.* [28] developed and launched the YOLO V6 algorithm. The authors focused on creating an object detector with an industry focus as a design strategy [30]. There are different configurations of YOLO V6 such as YOLO V6 n, YOLO V6 s, YOLO V6 m, YOLO V6 l and YOLO V6 l6 that allow adapting to different application scenarios. In that algorithm, the authors presented an updated reparametrized backbone and neck, proposed as the efficientRep backbone and the representation path aggregation network (Rep-PAN) neck ([9], [29]). Unlike YOLO V5, YOLO V6 features an anchor-free design [7] making it 51 percent quicker than anchor-based methods [29]. The reparametrized backbone with CSP and visual geometry group (VGG) backbones is used in the ‘‘m’’, ‘‘l’’ and ‘‘l6’’ variants, and ‘‘n’’ and ‘‘s’’ variants respectively [7]. The Neck of YOLO V6 is similar to YOLO V5, but breaking the convention, the head is efficiently decoupled, reducing computation and increasing precision by preventing parameter sharing between the detection and classification branches [7].

A two-loss function is required by YOLO V6. Distribution focal loss (DFL) [30] which allows to better learn where to place boxes, with more finesse and precision, by transforming on discrete distributions, regression into a classification task. And varifocal loss (VFL) [19] which is used as the classification loss, along with SIoU/GIoU (Scylla intersection over union/generalized intersection over union) [31] as regression loss [29].

Additional improvements of YOLO V6 focused on industrial applications include the use of knowledge distillation [32], involving a teacher-student model. The principle of this model is that a student model is trained by a teacher model because in order to train the student, the predictions of the teacher are used as soft labels along with the ground truth [29]. This comes without fueling the computational cost because based on [29], the aim is to replicate the powerful performance of the larger “teacher” model in a smaller “student” model.

3.7. YOLO V7

To implement YOLO V7, Wang *et al.* [33] built on previous versions of YOLO and significantly improved its architecture. The YOLO V7 model consists of seven variants: P6 models (d6, e6, w6, and e6e) and P5 models (v7, v7x, and v7-tiny) [7]. According to Yan *et al.* [11], the efficient layer aggregation network (E-LAN) represents the foundation of YOLO V7's architecture, which aims to optimize the inference speed [9] and set up a good network by controlling the longest and shortest gradients, enabling deeper networks to learn effectively and converge [11]. Wang *et al.* [33] also improved the architecture of YOLO V7 by using computational block in the YOLO V7’s backbone in order to introduce the extended efficient layer aggregation network (E-ELAN) which uses merge, shuffle and expand cardinality without disrupting the gradient path, in order to improve network learning [7].

YOLO V7 uses CSPDarkNet-53 backbone. To improve gradient flow during training, the CSPDarkNet-53 architecture consists of 53 convolutional layers with leaky rectified linear unit (LeakyReLU) activations and well-chosen filter sizes [9]. According to the same researchers, this backbone is enhanced with the module of SPP to extract multi-scale features crucial effectively for object detection. Regarding the head component, YOLO V7 uses the conventional shared-feature approach. [9] affirm that in this approach, before redirecting features to separate branches for class probability classification and bounding box coordinate regression, the characteristics are treated through a series of convolutional layers.

3.8. YOLO V8

In 2023, Ultralytics introduced YOLO V8 and proposed it in 5 versions (n, s, m, l, and x) [34]. It delivers some of the most advanced performance to date [35]. It is developed by PyTorch and offers real-time prediction of object bounding boxes and class probabilities using a fully convolutional network to provide a single-stage object detection model [36]. According to Yan *et al.* [11], it contains most of the improvements from previous versions of YOLO. YOLO V8 does not use anchors [9]. Anchor-free detection reduces the number of box predictions, which speeds up non-maximal suppression (NMS) [9]. The modified version of the CSPDarknet53 architecture, E-ELAN, forms the basis of the backbone of YOLO V8 and the efficient PyTorch implementation improves the speed of inference and training [11]. The YOLO-SPP-Boost backbone is introduced in YOLO V8. During the training process, the YOLO-SPP-Boost backbone integrates residual connections between convolutional layers to facilitate smooth gradient flow [9]. For efficient feature fusion, YOLO V8 uses a Focus-V3 neck. And for multi-scale feature extraction, this version of YOLO uses an SPP module [9].

The head of YOLO V8 adopts decoupled heads and breaks the conventional approach of YOLO V7 [9]. This involves that before making the final predictions, there must be a separation of features. According to [9], YOLO V8 further increases accuracy by using anchor-free design. That approach helps improving detection accuracy and flexibility [34]. It can directly predict the dimensions of the bounding box and the center point in order to simplify the network architecture while improving localization accuracy [9].

3.9. YOLO V9

In February 2024, Wang *et al.* [37] introduced YOLO V9. That version of YOLO, has improved its network architecture which is based on the robust codebase provided by Ultralytics YOLO V5 [38] in order to perform better in recognizing complex targets and with various sizes [39]. In addition to that, the enhanced model demonstrates significant improvements in multi-target scenarios and handling occlusions, which makes its use more reliable at complex intersections [39]. Relying on fewer calculations and parameters, YOLO V9 may have the same or better detection results than previous YOLO algorithms by extracting with precision and retaining information required to map data to targets [40].

YOLO V9 presents two key innovations: To improve parameter utilization efficiency, the first innovation of YOLO V9, represented by a new generalized efficient layer aggregation network (GELAN), is used [41]. The second innovation of YOLO V9 is a novel concept of programmable gradient information (PGI) framework [42] to propagate multi-level gradient information through an auxiliary reversible branch [41] in response to the problem of information loss in deep network data transmission [40]. Sharma *et al.* [36] affirm that those innovations can address issues related to computational efficiency effectively and information loss and according to Marchi *et al.* [39], it represents a significant leap forward in terms of efficiency, accuracy, and adaptability. Indeed, pursuing a train from scratch strategy, Wang *et al.* [37] obtained better detection results than state of the art models pre-trained with large datasets [41].

3.10. YOLO V10

Created by Wang *et al.* in 2024 [43], the YOLO V10 optimized the speed and accuracy of object localization in images and categorization compared to its earlier versions. There are different configurations of YOLO V10 such as YOLO V10n, YOLOV10 s, YOLO V10 m, YOLO V10b, YOLOV10l and YOLO V10x that allow adapting to different application scenarios [44]. YOLO V10 aims to further advance the performance-efficiency boundary of YOLOs from both the model architecture and the post-processing [43]. According to Sapkota *et al.* [44], YOLO V10 effectively addresses previous architectural limitations and the reliance on NMS, which is a significant step forward in enhancing inference speed, performance and operational efficiency. It provides two main improvements over previous versions of YOLO: a model design focused on accuracy and efficiency (to increase overall performance) and a consistent dual assignment (in order to have an NMS-free training protocol) [43]. The NMS-free training protocol through consistent dual assignments simplifies the output stage, reduces post processing time [44] and brings the competitive performance and low inference latency [43].

The architectural enhancements in YOLO V10 include the spatial-channel decoupled down sampling, the implementation of lightweight classification heads, and rank-guided block design, each contributing to substantial reductions in parameter count and computational demands [45]. YOLO V10 has improved its backbone by using an updated version of cross stage partial network (CSPNet) [44]. In order to have efficient and accurate feature extraction, this update is designed to reduce computational redundancy and improve gradient flow [44].

Regarding the neck of the architecture, it incorporated a path aggregation network to facilitate efficient multi-scale feature fusion [44]. This option is essential because it improves the algorithm's ability to detect objects with greater accuracy and for different sizes [44]. As for the head, YOLO V10 uses a dual-head design (one-to-many head and one-to-one head). One-to-Many head generates multiple predictions per object to improve learning accuracy by providing rich supervision signals [44]. One-to-One head delivers an optimal and single prediction per object. This eliminates the need for NMS and thus reduces latency and simplifies the detection process [44].

3.11. YOLO V11

Introduced by Ultralytics in 2024, YOLO V11 is the latest addition to the YOLO series algorithms building on the foundation of YOLO V8 to date [46] and was announced at the YOLO Vision 2024 conference [46]. There are different models of YOLO V11 with a scale that varies from small to large such as YOLO V11n, YOLO V11s, YOLO V11m, YOLO V11l and YOLO V11x that allow adapting to different application scenarios [46]. Optimizing performance across multiple computer vision tasks is one of the objectives of the training process of YOLO V11, just like for YOLO V10 [44]. This latest version of YOLO, brings solid improvements in the architecture and training methods that allow to bring even more speed, accuracy and efficiency [46]. To capture complex details in images, feature extraction capabilities have been optimized in the YOLO V11 architecture [44]. Indeed, according to Ultralytics [47], this version enhances speed and efficiency through training pipelines and optimized designs, balancing precision and performance and improves feature extraction with an advanced backbone and neck architecture YOLO V11 features the convolutional block with parallel spatial attention (C2PSA (cross stage partial with spatial attention)) components, spatial pyramid pooling fast (SPPF) and a cross-stage portion of kernel size 2 (C3k2) block, in order to improve the feature extraction capabilities of the model [46].

According to Sapkota *et al.* [44], YOLO V11 allows having better results on benchmark datasets because it uses enhanced training techniques. Indeed, on the COCO dataset, by using 22% fewer parameters compared to the YOLO V8m algorithm, YOLO V11m achieved a mAP score of 95% [44]. This demonstrates greater efficiency without compromising accuracy. Through an average inference speed 2% faster than YOLO V10, even in complex environments, YOLO V11 guarantees fast processing because it is optimized for real-time applications [44]. These data demonstrate that YOLO V11 represents a significant advancement in the field of artificial intelligence and particularly in areas where precision and rapid analysis are required.

4. RESULTS AND DISCUSSION

4.1. Comparison between the different versions of YOLO

This section contains the evolution of each version of YOLO including the architecture of each version. In addition to this we compare all the versions by presenting the advantages and disadvantages of each one. To be able to compare the 11 versions of the YOLO model, we applied it to the COCO dataset, which is a widely used dataset and includes 5000 images of everyday objects. For YOLO versions that have multiple variants, we have chosen to test the small(s) variant of all versions. The results of our comparison are as follows:

4.2. DISCUSSION

This literature review focuses on the YOLO algorithm; an artificial intelligence algorithm designed for object detection in images or videos. It provides a comprehensive and comparative analysis of the evolution of the YOLO algorithm from 2016 to 2024. While many papers focus on individual versions of YOLO or introduce specific changes or improvements, our work synthesizes all major improvements and advancements, highlights strengths and limitations of each version in order to help future researchers choosing the best version based on the needs of their studies.

This study aims to identify the advantages and disadvantages and present the contribution of each version of YOLO in terms of architecture, accuracy, speed, and improvements in order to know the best version to use for every requirement. This represents a decision-making support for researchers who are hesitating between different versions of the YOLO algorithm. The novel contribution of this manuscript is that the comparison made between the eleven versions of YOLO has not been previously consolidated in the literature especially a comparison that includes the latest versions 9, 10 and 11.

This literature review is based on the methodology of Kitchenham and Charters [3] aligned with PRISMA 2020 and the used techniques are as follows: 1/The formulation of the research questions was made based on the PICO technique as shown in Table 1, 2/for the search strategy, keywords/search strings use advanced Boolean queries on scientific databases, 3/selection process is based on the definition of inclusion and exclusion criteria, 4/ for document validity assessment, the used technique is critical assessment of the quality of resources based on quality assessment questions, 5/ finally, for data collection, we used structured extraction in Tables 2, 3 and 4.

The overview of this article answers the first question of this review by displaying the changes made to the YOLO algorithm in each of its versions. Table 2 allows us to know that YOLO V1 and YOLO V2 use an architecture without backbone, neck and head who are CNN and Darknet-19 respectively. In Table 3, we display the versions of YOLO algorithm that use an architecture with backbone, neck and head which are from Yolo V3 to Yolo v11 and that allow us to see the different architectures of each version.

Table 2. Architecture without Backbone, neck and head

YOLO	Year	Architecture
V1	2016	CNN
V2	2017	Darknet-19

Table 3. Architecture with Backbone, neck and head

YOLO	Year	Architecture with backbone, neck and head		
		Backbone	Neck	Head
V3	2018	Darknet-53	FPN, PANet	Anchor-based approaches
V4	2020	CSPDarknet-53	PANet, SPP	Anchor-based approaches
V5	2020	Modified CSPDarkNet53	PANet	Anchor-based approaches
V6	2022	EfficientRep	Rep-PAN	Anchor-free design
V7	2023	CSPDarkNet-53 enhanced with the module of SPP (E-ELAN)	PANet	Anchor-based approaches
V8	2023	E-ELAN	Focus-V3 neck	Decoupled heads, Anchor-free design
V9	2024	GELAN	RepNCSP-ELAN4*, **ADown** **	Anchor-free design
V10	2024	Updated version of CSPNet	PAN	Dual-head design (One-to-Many Head and One-to-One Head).
V11	2024	C3k2	SPFF***	Anchor-free design

* : RepNCSP-ELAN4: Reparametrized non-local cross stage partial efficient layer aggregation network
**: ADown; Attention downsampling
***: SPFF: Spatial pyramid fine fusion

Concerning the second question of this review, Table 4 presents the list of advantages and disadvantages of all versions of YOLO (V1 to V11) and Table 5 gives us a comparative analysis of the eleven versions on COCO dataset and this leads us to the following results. In terms of speed and accuracy, YOLO V2 is better than YOLO V1 but both algorithms are not suitable for complex contexts where objects are very close to each other because they can't detect them correctly. Comparing to YOLO V1 and V2, YOLO V3 improves the small-size target detection accuracy and prediction. As for YOLO V4, we can say that it is better than YOLO V3 in terms of precision and speed but according to [48] it may be slower in some scenarios. Concerning YOLO V5, it has 37.5% mAP for the small variant and it allows identifying very small objects faster than YOLO V4 but it encounters some difficulties when there is an overlap of objects, a change of lighting and other complex conditions.

YOLO V6 achieves high accuracy, performance and speed compared to previous versions but has not been much used because it is not fully open-source. As for YOLO V7, we can say that it proves to be effective in terms of accuracy compared to previous versions and in some scenarios, it even outperforms YOLO V8 but it still encounters problems in extreme conditions such as low lighting, or very cluttered backgrounds. YOLO V8 is user-friendly algorithm characterized by better speed and accuracy compared to previous algorithms except YOLO V7 which largely surpasses it in terms of accuracy in certain contexts. Our performance comparison charts as shown in Table 5 shows that YOLO V9s allows a significant reduction in the number of parameters used in the algorithm and compared to YOLO V8s, it has good accuracy 47%mAP but its speed remains relatively low. In addition to that, some authors claim that YOLO V9 achieves good performance and improves detection accuracy but other researchers find that YOLO V9 does not demonstrate outstanding performance in terms of speed and accuracy and still encounters problems in detecting low quality images. Concerning YOLO V10s, it has also a reduced number of parameters and has a good mAP (46,2%) and speed (137 FPS) but according to [34] its accuracy quality remains inferior to YOLO V8 for small objects and in some contexts it remains even inferior to YOLO V5. Finally, YOLO V11 offers increased efficiency and speed compared to previous versions but since it is a very recent version, it has not been tested by several researchers and in several contexts.

To answer the third and last question of this review, these results show that there is not a perfect algorithm. Choosing the appropriate version depends on the study context, the information used for the study, the complexity of the objects to be detected and priorities of researchers. Studies whose objective is the detection of small objects should use YOLO V4, YOLO V5, YOLO V6, YOLO V7, YOLO V8, YOLO V9, YOLO V10 or YOLO V11. If the priority of the study is the detection of small objects in environments that are complex, have low lighting or have very cluttered backgrounds, it is necessary to avoid working with YOLO V1, YOLO V2, YOLO V3, YOLO V5, YOLO V7 and YOLO V9. For studies where processing speed is a priority, they should use YOLO V5, YOLO V6, YOLO V7, YOLO V8, YOLO V10 or YOLO V11 and avoid working with YOLO V4 or YOLO V9.

Table 4. Advantages and disadvantages of the different versions of YOLO algorithm

Version of YOLO	Advantages	Disadvantages
YOLO V1	1/ For candidate box prediction and object classification, YOLO V1 directly adopts regression. This improvement makes YOLO V1 ten times faster in terms of detection than faster R-CNN (regions with CNN) and simplifies the network structure [46]	1/ In scenarios where objects are close to each other, the performance of YOLO V1 is not optimal [9]
YOLO V2	1/ Faster in terms of detection speed than other detection algorithms at that time, including YOLO V1 ([13], [46]) 2/Furthermore, it can be run at a variety of image sizes to provide a smooth tradeoff between accuracy and speed [13]	1/YOLO V2 cannot effectively detect small objects and very close objects in complex images [13]
YOLO V3	1/More effective than YOLO V2 [46] 2/ Has a low background false detection rate and very fast detection speed, and improves the detection accuracy of small objects [46] 3/ The most significant improvement in YOLO V3 lies in multi-scale predictions [8]	1/ Poor in terms of its prediction accuracy of target coordinates [46]
YOLO V4	1/ Compared to YOLO V3, YOLO V4 achieves a higher frames per second (FPS) and average precision (AP) rate [8]	1/YOLO V4 may be slower [48]
YOLO V5	1/YOLO V5 enables the model to learn how to identify a variety of objects at a much smaller scale than normal [8]	1/YOLO V5 may not be robust enough under complex conditions, such as varying distances, object overlap, occlusions, and lighting changes [49]
YOLO V6	1/YOLO V6 achieves higher average precision (mAP) on standard datasets like COCO [29] 2/It combines advanced techniques like RepOptimizer and structural optimizations to improve object detection performance [28].	1/Compared to previous versions like YOLO V4 or YOLO V5, YOLO V6 has been less used by the research community 2/YOLO V6 is not fully open-source in some cases, which may be problematic for small companies or independent researchers.
YOLO V7	1/ Through several means, YOLO V7 improves accuracy, notably through model scaling and parameter tuning and also through the use of E-ELAN [8] 2/ Recent work has shown that YOLO V5, YOLO V6, and even YOLO V8, in some scenarios, are outperformed by YOLO V7 ([8], [50]) 3/In terms of resource efficiency and familiarity, YOLO V7 proves to be quite effective [9]	1/Although the YOLO V7 is designed to perform well in complex field environments, it may still encounter extreme conditions such as poor lighting, heavy occlusion, or very cluttered backgrounds [51]
YOLO V8	1/The advantages of YOLO V8 are scalability, high performance, and user-friendliness [34] 2/ YOLO V8 offers superior accuracy and speed compared to previous YOLO versions [42] 3/YOLO V8 is enthusiastic about its cutting-edge design, which maximizes uniqueness and functionality [9]	1/YOLO V8 is outperformed by YOLO V7 [50]
YOLO V9	1/YOLO V9 achieved good performance [52] and introduces new architectures that improve accuracy (e.g., generalized high-efficiency layer aggregation networks and programmable gradient information) [42] 2/ YOLO V9 allows learning effectively and identifying different types of objects in real time and process large-scale data sets [40]	1/ Compared to YOLO V8, YOLO V9 fails to find the right balance between accuracy and speed [42] 2/Detecting low-quality image targets by YOLO V9 still poses challenges [52] 3/ YOLO V9's processing speed and accuracy still needs improvement ([53], [39])
YOLO V10	1/ By optimizing model components and removing NMS, YOLO V10 excels in lightweight, accuracy, and speed [34]	1/ In detecting small targets, YOLO V10 is slightly inferior to YOLO V8 [34] 2/According to Geetha and Hussain [54], in certain scenarios, YOLO V5 outperformed YOLO V10 and YOLO V8 in terms of accuracy
YOLO V11	1/YOLO V11 allows a significant reduction in the number of parameters used in the algorithm [47] 2/YOLO V11 offers increased efficiency and speed compared to previous versions [47]	1/Since it's a very recent version; YOLO V11 has not been tested and optimized over a long period.

The study's findings and results demonstrate that the eleven versions of the YOLO algorithm have undergone continuous improvement over the years in terms of accuracy, architectural complexity, and object detection speed. Indeed, the first three versions established the basics of detection, but they are not very accurate and struggle to detect objects in complex environments. Versions 4 to 11 allow the detection of small objects but they do not all perform well in complex environments. Indeed, YOLO V5, YOLO V7 and YOLO V9 do not perform well in the environments that are not very clear or lack brightness. The most recent versions V10 and V11 offer a fairly balanced compromise between speed, precision and number of parameters and they are especially very efficient in terms of speed. This is due to innovations such as decoupled heads, anchor-free detection and transformer-based modules.

The comparison between different versions of YOLO shows that recent versions are not always the best in all contexts. For example, YOLO V4 may demonstrate excellent performance for small object detection, while YOLO V8 prioritizes speed and sacrifices some accuracy. This shows that the best version of the YOLO algorithm depends on the context and that researchers should not always choose the latest version, as it may not yield the desired results in all contexts. The architectures adopted for each version are the result of good balance that developers consider based on the constraints imposed by the study.

Table 5. Comparative performance charts

YOLO	mAP50-95*	FPS **	Parameters (M)***
YOLO V1	19.6%	45	62 M
YOLO V2	21.6%	67	50 M
YOLO V3	31.7%	30	61.9 M
YOLO V4	44,5%	62	64 M
YOLO V5s	37.5%	132	7.2 M
YOLO V6s	44.8%	484	13.14 M
YOLO V7	51.4%	135	37.2 M
YOLO V8s	44.9%	133	11.16 M
YOLO V9s	47%	120	7.20 M
YOLO V10s	46.2%	137	7.25 M
YOLO V11s	46.8%	149	9.44 M

*** : mAP:** mean Average Precision determined for IoU levels between 0.50 and 0.95 in steps of 0.05, then averaged over all classes

**** : FPS :** Frames Per Second

***: **Parameters (M):** Number of parameters of the model

5. CONCLUSION

Based on our results and discussion, this literature review offers a roadmap for researchers to choose the YOLO version best suited to their priorities, whether in terms of accuracy, performance, or response time. The implications of this work include rapid algorithm selection based on specific constraints of a study and helping researchers avoiding errors and time-consuming when choosing a model. In addition to that, this review will allow researchers having efficient and improved performance in the use of artificial intelligence. As an outlook for this work, we recommend that future research focus on exploring models that achieve a good balance between accuracy, speed and number of parameters to improve detection efficiency, especially for uses where real-time detection is necessary.

FUNDING INFORMATION

No funding involved.

AUTHOR CONTRIBUTIONS STATEMENT

This journal uses the Contributor Roles Taxonomy (CRediT) to recognize individual author contributions, reduce authorship disputes, and facilitate collaboration.

Name of Author	C	M	So	Va	Fo	I	R	D	O	E	Vi	Su	P	Fu
Imane Beqqali Hassani	✓	✓	✓					✓	✓	✓				
Soufia Benhida					✓									
Nabil Lamii												✓		
Khalid Oqaidi				✓			✓							
Ahmed Ouiddad													✓	
Soukaina Ghiadi													✓	

C : **C**onceptualization
M : **M**ethodology
So : **So**ftware
Va : **Va**lidation
Fo : **Fo**rmal analysis
I : **I**nvestigation
R : **R**esources
D : **D**ata Curation
O : Writing - **O**riginal Draft
E : Writing - Review & **E**diting
Vi : **Vi**sualization
Su : **Su**pervision
P : **P**roject administration
Fu : **Fu**nding acquisition

CONFLICT OF INTEREST STATEMENT

Authors state no conflict of interest.

DATA AVAILABILITY

The authors confirm that the data supporting the findings of this study are available within the article

REFERENCES

[1] B. Li, L. Liu, H. Jia, Z. Zang, Z. Fu, and J. Xi, "YOLO-TP: A lightweight model for individual counting of Lasioderma serricorne," *Journal of Stored Products Research*, vol. 109, p. 102456, Dec. 2024, doi: 10.1016/j.jspr.2024.102456.

[2] N. M. Alahdal, F. Abukhodair, L. H. Meftah, and A. Cherif, "Real-time object detection in autonomous vehicles with YOLO," *Procedia Computer Science*, vol. 246, no. C, pp. 2792–2801, 2024, doi: 10.1016/j.procs.2024.09.392.

[3] B. Kitchenham and S. M. Charters, "Guidelines for performing systematic literature reviews in software engineering," UK, 2007.

[4] M. J. Page *et al.*, "The prisma 2020 statement: An updated guideline for reporting systematic reviews," *Medicina Fluminensis*, vol. 57, no. 4, pp. 444–465, 2021, doi: 10.21860/medflum2021_264903.

[5] W. S. Richardson, M. C. Wilson, J. Nishikawa, and R. S. Hayward, "The well-built clinical question: a key to evidence-based decisions.," *ACP journal club*, vol. 123, no. 3, p. A12, Nov. 1995, doi: 10.7326/acpjc-1995-123-3-a12.

[6] C. Gheorghe, M. Duguleana, R. G. Boboc, and C. C. Postelnicu, "Analyzing real-time object detection with YOLO algorithm in automotive applications: a review," *CMES - Computer Modeling in Engineering and Sciences*, vol. 141, no. 3, pp. 1939–1981, 2024, doi: 10.32604/cmes.2024.054735.

[7] A. Ahmed, A. S. Imran, A. Manaf, Z. Kastrati, and S. M. Daudpota, "Enhancing wrist abnormality detection with YOLO: Analysis of state-of-the-art single-stage detection models," *Biomedical Signal Processing and Control*, vol. 93, p. 106144, Jul. 2024, doi: 10.1016/j.bspc.2024.106144.

[8] J. Doherty, B. Gardiner, E. Kerr, and N. Siddique, "BiFPN-YOLO: One-stage object detection integrating Bi-directional feature pyramid networks," *Pattern Recognition*, vol. 160, p. 111209, Apr. 2025, doi: 10.1016/j.patcog.2024.111209.

[9] F. Fahim and M. S. Hasan, "Enhancing the reliability of power grids: A YOLO based approach for insulator defect detection," *e-Prime - Advances in Electrical Engineering, Electronics and Energy*, vol. 9, p. 100663, Sep. 2024, doi: 10.1016/j.prime.2024.100663.

[10] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You Only Look Once: Unified, real-time object detection," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Jun. 2016, vol. 2016-Decem, pp. 779–788, doi: 10.1109/CVPR.2016.91.

[11] J. Yan *et al.*, "Enhanced object detection in pediatric bronchoscopy images using YOLO-based algorithms with CBAM attention mechanism," *Heliyon*, vol. 10, no. 12, p. e32678, Jun. 2024, doi: 10.1016/j.heliyon.2024.e32678.

[12] C. Szegedy *et al.*, "Going deeper with convolutions," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Jun. 2015, vol. 07-12-June-2015, pp. 1–9, doi: 10.1109/CVPR.2015.7298594.

[13] J. Redmon and A. Farhadi, "YOLO9000: Better, faster, stronger," in *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, Jul. 2017, vol. 2017-January, pp. 6517–6525, doi: 10.1109/CVPR.2017.690.

[14] X. Han, J. Chang, and K. Wang, "Real-time object detection based on YOLO-v2 for tiny vehicle object," *Procedia Computer Science*, vol. 183, pp. 61–72, 2021, doi: 10.1016/j.procs.2021.02.031.

[15] W. Liu *et al.*, "Multi-objective performance evaluation of the detection of catenary support components using DCNNs," *IFAC-PapersOnLine*, vol. 51, no. 9, pp. 98–105, 2018, doi: 10.1016/j.ifacol.2018.07.017.

[16] J. Redmon and A. Farhadi, "YOLOv3: An incremental improvement," *arXiv preprint arXiv:1804.02767*, 2018.

[17] H. Gu, K. Zhu, A. Strauss, Y. Shi, D. Sumarac, and M. Cao, "Rapid and accurate identification of concrete surface cracks via a lightweight & efficient YOLOv3 algorithm," *SDHM Structural Durability and Health Monitoring*, vol. 18, no. 4, pp. 377–394, 2024, doi: 10.32604/sdhm.2024.042388.

[18] J. Redmon, "Darknet: open source neural networks in C." http://pjreddie.com/darknet/ (accessed Apr. 01, 2025).

[19] M. Zhang, H. Liang, Z. Wang, L. Wang, C. Huang, and X. Luo, "Damaged apple detection with a hybrid YOLOv3 algorithm," *Information Processing in Agriculture*, vol. 11, no. 2, pp. 163–171, Jun. 2024, doi: 10.1016/j.inpa.2022.12.001.

[20] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, "YOLOv4: Optimal speed and accuracy of object detection," *arXiv preprint arXiv:2004.10934*, 2020.

[21] C. Y. Wang, H. Y. Mark Liao, Y. H. Wu, P. Y. Chen, J. W. Hsieh, and I. H. Yeh, "CSPNet: A new Backbone that can enhance learning capability of CNN," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, 2020, vol. 2020-June, pp. 1571–1580, doi: 10.1109/CVPRW50498.2020.00203.

[22] S. Liu, L. Qi, H. Qin, J. Shi, and J. Jia, "Path aggregation network for instance segmentation," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2018, pp. 8759–8768, doi: 10.1109/CVPR.2018.00913.

[23] K. He, X. Zhang, S. Ren, and J. Sun, "Spatial pyramid pooling in deep convolutional networks for visual recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 9, pp. 1904–1916, Sep. 2015, doi: 10.1109/TPAMI.2015.2389824.

[24] M. Xu, S. Yoon, A. Fuentes, and D. S. Park, "A comprehensive survey of image augmentation techniques for deep learning," *Pattern Recognition*, vol. 137, May 2023, doi: 10.1016/j.patcog.2023.109347.

[25] G. Jocher, "YOLOv5: GitHub repository," *GitHub*. 2020. [Online]. Available: https://github.com/ultralytics/yolov5

[26] J. Nelson, "YOLOv5 is here." [Online]. Available: https://blog.roboflow.com/yolov5-is-here/

[27] G. Jocher, "6.2-YOLOv5 classification models: Apple M1, reproducibility, ClearML and deci.ai integrations." 2022, doi: 10.5281/zenodo.7347926.

[28] C. Li *et al.*, "YOLOv6: A single-stage object detection framework for industrial applications," *arXiv preprint arXiv:2209.02976*, Sep. 2022.

[29] M. Hussain, "YOLO-v1 to YOLO-v8, the rise of YOLO and its complementary nature toward digital manufacturing and industrial defect detection," *Machines*, vol. 11, no. 7, p. 677, Jun. 2023, doi: 10.3390/machines11070677.

[30] X. Li *et al.*, "Generalized focal loss: Learning qualified and distributed bounding boxes for dense object detection," in *Advances*

in Neural Information Processing Systems, 2020, vol. 2020-December.
[31] Z. Gevorgyan, "SIoU loss: More powerful learning for bounding box regression," *arXiv preprint*, 2022, doi: 10.48550/arXiv.2205.12740.
[32] C. Shu, Y. Liu, J. Gao, Z. Yan, and C. Shen, "Channel-wise knowledge distillation for dense prediction," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 5311–5320.
[33] C. Y. Wang, A. Bochkovskiy, and H. Y. M. Liao, "YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Jun. 2023, vol. 2023-June, pp. 7464–7475, doi: 10.1109/CVPR52729.2023.00721.
[34] L. Chen, G. Li, S. Zhang, W. Mao, and M. Zhang, "YOLO-SAG: An improved wildlife object detection algorithm based on YOLOv8n," *Ecological Informatics*, vol. 83, p. 102791, Nov. 2024, doi: 10.1016/j.ecoinf.2024.102791.
[35] J. Terven, D. M. Córdova-Esparza, and J. A. Romero-González, "A comprehensive review of YOLO architectures in computer vision: From YOLOv1 to YOLOv8 and YOLO-NAS," *Machine Learning and Knowledge Extraction*, vol. 5, no. 4, pp. 1680–1716, Nov. 2023, doi: 10.3390/make5040083.
[36] A. Sharma, V. Kumar, and L. Longchamps, "Comparative performance of YOLOv8, YOLOv9, YOLOv10, YOLOv11 and Faster R-CNN models for detection of multiple weed species," *Smart Agricultural Technology*, vol. 9, p. 100648, Dec. 2024, doi: 10.1016/j.atech.2024.100648.
[37] C. Y. Wang, I. H. Yeh, and H. Y. Mark Liao, "YOLOv9: Learning what you want to learn using programmable gradient information," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2025, vol. 15089 LNCS, pp. 1–21, doi: 10.1007/978-3-031-72751-1_1.
[38] R. Marchi, S. Hau, K. M. Suryaningrum, and R. Yunanda, "Comparing YOLOv8 and YOLOv9 algorithm on breast cancer detection case," *Procedia Computer Science*, vol. 245, no. C, pp. 239–246, 2024, doi: 10.1016/j.procs.2024.10.248.
[39] R. An, X. Zhang, M. Sun, and G. Wang, "GC-YOLOv9: Innovative smart city traffic monitoring solution," *Alexandria Engineering Journal*, vol. 106, pp. 277–287, Nov. 2024, doi: 10.1016/j.aej.2024.07.004.
[40] W. Pan, Y. Wang, B. Lv, L. Wang, and L. Li, "Research on multi-UUV mission planning for mine countermeasures based on preferred multi-objective optimization," *Ain Shams Engineering Journal*, vol. 15, no. 11, p. 103006, Nov. 2024, doi: 10.1016/j.asej.2024.103006.
[41] B. Deng, Y. Lu, and Z. Li, "Detection, counting, and maturity assessment of blueberries in canopy images using YOLOv8 and YOLOv9," *Smart Agricultural Technology*, vol. 9, p. 100620, Dec. 2024, doi: 10.1016/j.atech.2024.100620.
[42] M. Hussain and R. Khanam, "In-depth review of YOLOv1 to YOLOv10 variants for enhanced photovoltaic defect detection," *Solar*, vol. 4, no. 3, pp. 351–386, Sep. 2024, doi: 10.3390/solar4030016.
[43] A. Wang and Others, "YOLOv10: Real-time end-to-end object detection," in *Conference on Neural Information Processing Systems*, 2024, pp. 107984–108011.
[44] R. Sapkota, Z. Meng, and M. Karkee, "Synthetic meets authentic: Leveraging LLM generated datasets for YOLO11 and YOLOv10-based apple detection through machine vision sensors," *Smart Agricultural Technology*, vol. 9, p. 100614, Dec. 2024, doi: 10.1016/j.atech.2024.100614.
[45] M. A. R. Alif and M. Hussain, "YOLOv1 to YOLOv10: A comprehensive review of YOLO variants and their application in medical image detection," *Journal of Artificial Intelligence Practice*, vol. 7, no. 3. 2024, doi: 10.23977/jaip.2024.070314.
[46] D. Wang *et al.*, "SDS-YOLO: An improved vibratory position detection algorithm based on YOLOv11," *Measurement: Journal of the International Measurement Confederation*, vol. 244, p. 116518, Feb. 2025, doi: 10.1016/j.measurement.2024.116518.
[47] Ultralytics, "YOLO11 new," *Ultralytics.Com*, 2024. https://docs.ultralytics.com/models/yolo11/ (accessed Apr. 01, 2025).
[48] I. Haq *et al.*, "YOLO and residual network for colorectal cancer cell detection and counting," *Heliyon*, vol. 10, no. 2, Jan. 2024, doi: 10.1016/j.heliyon.2024.e24403.
[49] R. Yang, Y. He, Z. Hu, R. Gao, and H. Yang, "CA-YOLOv5: A YOLO model for apple detection in the natural environment," *Systems Science and Control Engineering*, vol. 12, no. 1, 2024, doi: 10.1080/21642583.2023.2278905.
[50] M. U. Rehman, H. Eesaar, Z. Abbas, L. Seneviratne, I. Hussain, and K. T. Chong, "Advanced drone-based weed detection using feature-enriched deep learning approach," *Knowledge-Based Systems*, vol. 305, p. 112655, Dec. 2024, doi: 10.1016/j.knosys.2024.112655.
[51] J. Chappidi and D. M. Sundaram, "Enhanced animal detection in complex outdoor environments using modified YOLO V7," *International Journal of Intelligent Systems and Applications in Engineering*, vol. 12, no. 19s, pp. 375–382, 2024.
[52] N. Kang, F. Ma, W. Wan, D. Wang, H. Yao, and K. Sheng, "Improved YOLOv9-based objects detection in adverse weather conditions for autonomous driving," *IFAC-PapersOnLine*, vol. 58, no. 29, pp. 267–271, 2024, doi: 10.1016/j.ifacol.2024.11.155.
[53] M. Feng *et al.*, "A computer vision and RFID fusion-based method for measuring individual feed intake and its application for detecting individual differences in feed efficiency of large yellow croaker (Larimichthys crocea)," *Water Biology and Security*, vol. 4, no. 2, p. 100332, Apr. 2025, doi: 10.1016/j.watbs.2024.100332.
[54] A. S. Geetha and M. Hussain, "A comparative analysis of YOLOv5, YOLOv8, and YOLOv10 in kitchen safety," *arXiv preprint*, 2024, doi: 10.48550/arXiv.2407.20872.

BIOGRAPHIES OF AUTHORS

Imane Beqqali Hassani is an associate professor at the Department of Research, Development, and Innovation Laboratory, Mundiapolis University, Casablanca, Morocco, where she has been a faculty member since 2020. From 2011 to 2013, she worked at E-business consulting solutions as an IT consultant, and then she began teaching as part-time work in several higher education establishments. Her passion for teaching led her to do her Ph.D. and she obtained it in 2020. From 2021 to 2025, she was also the head of the applied computer science degree program. Her research interests are primarily in the area of computer science and especially in computer vision. She can be contacted at email: i.beqqali@gmail.com.

Soufia Benhida was born in 1989 in Meknès, Morocco, she received her Technical University Diploma in Digital Electronics (DUT) in 2009, then she got the Engineer Diploma in Industrial Engineering (DIE) in 2012. Following this, in 2018, Benhida Soufia earned her Ph.D. in mathematics from the University of Normandy in Rouen at the Mathematical Laboratory of the National Institute of Applied Sciences of Rouen, France (LMI), and her Ph.D. in Applied Mathematics at the National School of Applied Sciences in Agadir Morocco, currently holding the position of researcher professor in mathematics and Head of the preparatory classes at the Engineering school at the Mundiapolis University of Casablanca. She can be contacted at email: s.benhida@gmail.com.

Lamii Nabil is a Ph.D. in Industrial Engineering and Logistics. In 2017, he started his academic career as a part-time lecturer in the fields of industrial engineering and artificial intelligence. Between 2017 and 2022, he organized several research events, including the CIMOCL – International Conference on Operational Management and Supply Chain. In recognition of his research efforts, he was awarded the First Doctoral Prize for the Best PhD Thesis in 2022/2023 by the Faculty of Sciences and Technology of Settat (FST Settat). His research focuses on the application of mathematical models and artificial intelligence tools to improve and optimize supply chain processes. He works particularly on multi-criteria decision-making methods and metaheuristic algorithms to enhance deep learning models applied in industrial and logistics contexts. Since 2024, he has been the Head of the Industrial Engineering Program at Mundiapolis University, where he has led several projects, including the accreditation of the program. He is also the Project Leader of the Intelligent Employability Initiative at Mundiapolis University. He can be contacted at email: n.lamii@mundiapolis.ma.

Khalid Oqaidi is a professor of computer science and artificial intelligence at Mundiapolis University. He holds a Ph.D. in Computer Science and Artificial Intelligence from the Faculty of Science and Technology (FST) of Mohammedia, Hassan II University of Casablanca. His research focuses on using data science techniques to improve the quality of higher education. He is also a telecommunications engineer, graduated from the National Institute of Posts and Telecommunications (INPT Rabat). Khalid is the founder and CEO of Entscheider, a consulting firm specializing in statistical data analysis, predictive modeling using machine learning and data science techniques, and market research. He can be contacted at email: khalid.oqaidi@gmail.com.

Ahmed Ouiddad received the M.Sc. degree in information systems engineering from Cadi Ayyad University, Marrakech, in 2010, and the Ph.D. degree in Computer Science and Information Systems from Hassan 1st University, Settat, Morocco, in 2021. Since 2012, he has been serving as a full-time professor of Information Systems at Sup de Co Marrakech. He is currently an affiliated researcher with the LAMSAD Laboratory at ENSA Berrechid and a part-time lecturer at several Moroccan academic institutions, including ENCG Settat, EMSI Marrakech, and FST Settat. Dr. OUIDDAD has authored and coauthored numerous peer-reviewed journal and conference papers indexed in Scopus and Web of Science. His research interests span enterprise resource planning (ERP) systems, decision support systems, digital transformation, and user adoption of information technologies. In addition to his academic work, he is an experienced ERP consultant and digital strategy expert. He can be contacted at email: ouiddad.ahmed@gmail.com.

Soukaina Ghiadi holds a Ph.D. in management sciences from the National School of Applied Sciences (ENSA) of Berrechid, Morocco. She also earned a master's degree in marketing and Commercial Action from Hassan First University, Settat, Morocco. She has served as a part-time lecturer at various Moroccan institutions, including ENSA Berrechid and the Faculty of Legal, Economic and Social Sciences of Ain Sebaa in Casablanca, Morocco. She also held the position of communication and marketing manager at TGCC Group in Casablanca. Her research interests include marketing metrics, marketing performance and machine learning. She can be contacted at email: gh.soukaina@gmail.com.