Automated bacteria and fungi classification using convolutional neural network on embedded system

Tarik Bouganssa[1], Maryem Ait Moulay[2], Samar Aarabi[3], Abdelali Lasfar[2], Abdellatif El Afia[1]
[1]Laboratory Smart System, ENSIAS, Mohammed V University, Rabat, Morocco
[2]Laboratory Systems Analysis and Information Processing and Industrial Management, High School of Technology SALE, Mohammadia School of Engineering, Mohammed V University, Rabat, Morocco
[3]Materials, Energy and Acoustics Team, High School of Technology SALE, Mohammed V University, Rabat, Morocco

Article Info

Article history:

Received Aug 12, 2024
Revised Jan 7, 2026
Accepted Jan 25, 2026

Keywords:

Bacteria
Convolutional neural network
Embedded system
Microbiology
YOLOv8

ABSTRACT

In this study, we created and applied novel concepts for hardware-based image identification and categorization. For artificial intelligence (AI) and image recognition applications, this includes putting algorithms for recognizing colors, textures, and shapes into practice. Our contribution uses an embedded device with a camera and a microcomputer (Raspberry-Pi4 type) to replace the optical assessment of Petri dishes. Our object recognition system processes images efficiently by using a state-of-the-art kernel function and a new neighborhood architecture. Using the well-known convolutional neural network (CNN) architecture, YOLOv8, as a pre-trained model, we evaluated the proposed CNN-based method for object recognition in a number of demanding scenarios. Several Petri plates, uncontrolled settings, and different backgrounds and illumination were used to evaluate the technology. Our dynamic mode integrates a CNN network with an attention mask to highlight the traits of bacteria and fungi, ensuring robust recognition. We implemented our algorithm on a Raspberry Pi 400, connected to a CMOS 3.0 camera sensor and a human-machine interface (HMI) for instant display of results.

This is an open access article under the CC BY-SA license.

Corresponding Author:

Tarik Bouganssa
Laboratory Smart System, ENSIAS, Mohammed V University
Rabat, Morocco
Email: tarik.bouganssa@gmail.com

1. INTRODUCTION

Bacterial infections pose an important danger to microbiology labs and healthcare institutions worldwide. Consequently, timely and precise identification of bacteria and fungi is essential for diagnosing and appropriately managing diseases [1]. This detection has historically been done manually, which takes time and is ineffective. Petri dishes are widespread and effective tools commonly used by researchers to collect data on samples during microbiological studies. They are generally equipped with a lump lens to clarify vision, the Petri dishes illustrated in Figure 1 deployed in microbiology laboratories have innovated in the field of research on bacteria and fungi [2].

In this context of object detection object specially bacteria and fungi detection, there are two types of convolutional neural network (CNN) algorithms designed to accelerate object detection models and achieve high accuracy region-based convolutional neural networks (R-CNN) and you only look once (YOLO) [3]. In this field, many studies have been published [4], to identify and quantify tilapia larvae using the Faster R-CNN R50-FPN 2X and Grid R-CNN-X101-32X4d-FPN 2X models gave the greatest outcomes, with a mean accuracy 50 of 97.30%. and [5] uses a digital twin-based intelligent health system with a cascade recurrent convolutional neural network architecture to identify COVID-19 from X-ray images. The model achieved an average accuracy of 0.94. Moreover, in several research projects like Ahmed *et al.* [6] uses multi-scale perceptual (MSP)-YOLO for the diagnosis of vaginitis and achieves sensitivities of 0.706 for key cells and 0.910 for *Trichomonas*, outperforming the reference model by 0.218 and 0.051. Also, Chen *et al.* [7] applied deformable convolution network (DCN)-YOLOv5, with a DCN, can detect objects, key points and track the *Oplegnathus punctatus*'s actions in an ammonia-rich environment. This model showed improved performance over the original YOLOv5, achieving mAP@0.5 of 93.71% and mAP@0.5:0.95 of 57.45%. In the same context of object detection, Jubayer *et al.* [8] applied YOLOv5 for detecting mold on food surfaces and YOLOv5 performed exceptionally well, achieving 98.10% accuracy, 100% recall, and 99.60% average precision (AP), outperforming versions YOLOv3 and YOLOv4.

Our contribution involves replacing the visual inspection of Petri dishes with an onboard system that uses the YOLOv8 model for object detection. This system employs a camera and a processing card in the form of a microcomputer to process images ibn real-time and automate the inspection. This approach not only minimizes human interference during data collection but also provides valuable visual data that researchers can use to better understand bacterial evolution in samples [9].

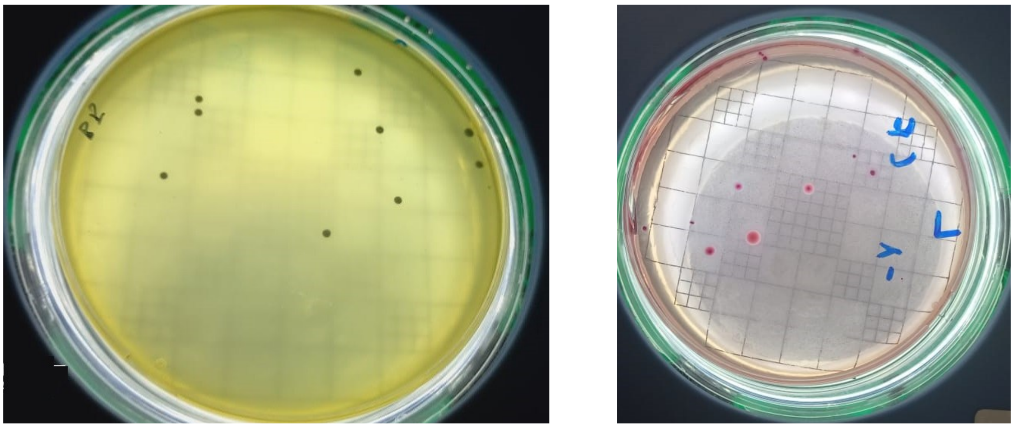


Figure 1. Petri dishes

2. THE IMPORTANCE OF PETRI DISHES IN THE LABORATORY

Petri dishes are superficial and cylindrical containers made of glass or plastic that are used to grow bacteria, fungi, and other microorganisms on specific nutrient media [10]. They are necessary in microbiology laboratories for producing and studying microorganisms. Offering a controlled environment that minimizes contamination [11].

Petri dishes have grown to be among the most frequently utilized lab supplies because of its simplicity and usability [12] a Petri dish is made up of a translucent base and a square or circular, loose-fitting cover that is intended to shield samples from outside contamination. Additionally, they are essential to the process of assessing antibiotic susceptibility, which establishes how well antibiotics work against different strains of bacteria [13]. Also, Landis *et al.* [14] demonstrates the benefits of the large surface area to volume ratio of Petri plates, which can help organisms such as *Brettanomyces bruxellensis* grow and produce more metabolites under aerobic circumstances.

To find new diseases and help microbiologists in the laboratory precisely count and analyze microbial content. It is critical to identify the bacteria and fungi growing on Petri dishes. To do this, we have implemented a model that makes use of an extensive dataset to enhance the identification and categorization of microorganisms, hence facilitating more accurate and insightful microbiological evaluations.

3. MATERIAL AND METHODS

3.1. Computer vision and convolutional neural networks

Artificial intelligence (AI) [15] was first described in 1950 [16]. Machine learning, a portion of AI, allows machines to acquire knowledge from data and advance without the need for explicit programming [17]. One subfield of machine learning called "deep learning (DL)" models difficult data using deep neural networks, due to its ability to identify objects in images, DL architectures have attracted a lot of attention.

As a result, they have been applied to medical imaging to perform tasks such as organ detection and segmentation [17]. A specific DL architecture created to effectively process visual data is the CNN.

Computer vision is another subset of AI that allows computers to see and understand [18] by training them to identify and interpret the content of images and videos. Since they are closely related to each other, advances in DL in recent years have also led to exceptional successes in the field of computer vision. Applications for computer vision include real-time sports, autonomous cars, object identification, and facial recognition.

CNNs [19], which were first introduced in the LeNet-5 architecture [20] by Yann LeCun *et al.* in 1998, received much attention after the release of Alexnet in 2012 [3]. With the availability of large datasets, CNNs are able to automatically detect important features, make highly accurate predictions, and perform computer vision tasks that were previously impossible. Unlike the classic neural network with fully connected layers, CNN has a unique architecture as shown in Figure 2 that generally includes three types of layers: convolutional layer, pooling layer, and fully connected layer.

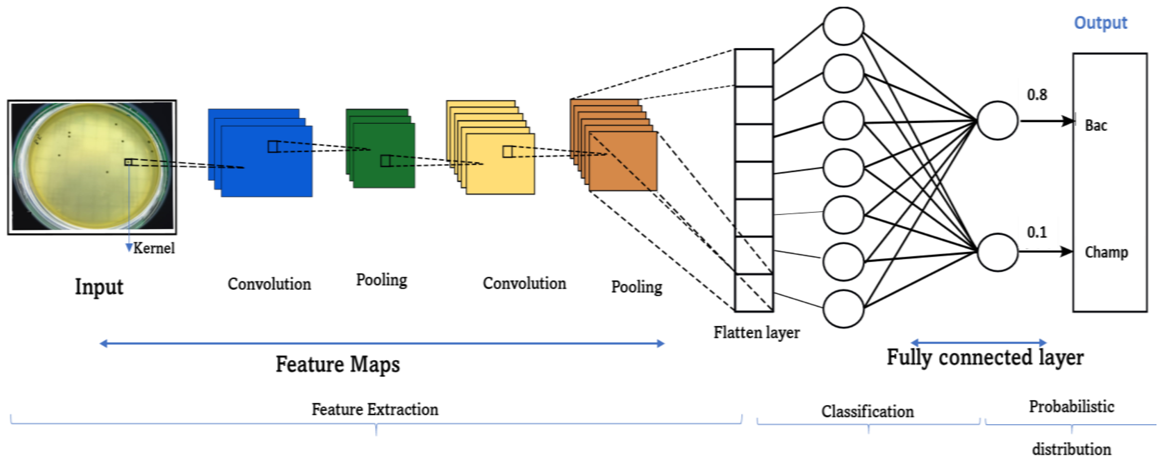


Figure 2. Schematic illustration of a CNN architecture

The YOLO model family was proposed by Redmon *et al.* [21] in 2015, as a state-of-the-art real-time object detection system. It is an object recognition and localization algorithm based on a deep neural network. It is best feature is that it works very fast. For example, if you enter an image, the system will display the objects it contains and the position of each object (the rectangular frame containing the object) [22]. YOLO is an outstanding accomplishment in object detection, albeit restricted to a single detection within a picture. Afterwards, numerous further iterations, ranging from YOLOv1 to YOLOv7 [23], were introduced with advancements in segmentation, multi-object identification in a single frame, accuracy, and exact localization [24]. The model was improved with batch normalization, anchor boxes, and dimension clusters in YOLOv2, which was released in 2016. YOLOv3, which was released in 2018, enhanced performance by employing spatial pyramid pooling and a more effective backbone network. With the release of YOLOv4, which debuted in 2020, additional features like anchor-free detecting head and mosaic data augmentation were added. With integrated experiment tracking and hyperparameter adjustment, YOLOv5 improved the model even further. Autonomous delivery robots employ YOLOv6 [25], which was made open-source in 2022, while YOLOv7 introduced pose estimation features. The most recent Ultralytics version, YOLOv8 [26], provides improved performance and adaptability for applications related to tracking, segmentation, and detection. YOLOv9 [27] presents programmable gradient information (PGI) and the generalized efficient layer aggregation network (GELAN), the Figure 3 illustrates various versions of YOLO algorithms.

YOLO works in that we take an image and divide it into an S×S grid, in each of the grids we take N bounding boxes. For each of the bounding boxes, the network generates a class probability and offset values for the bounding box. Bounding box prediction: Anchor boxes are made of dimension clusters by the YOLO algorithm to anticipate bounding boxes. For every bounding box, it is network predicts four coordinates: tx, ty, tw, and th. The predictions match Figure 4 if the cell is offset from the upper left corner of the image by (cx, cy) and the width and height of the previous bounding box are pw, ph.

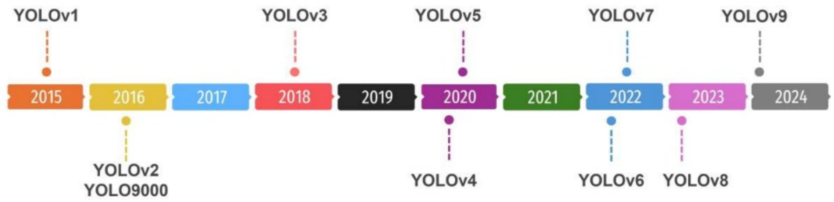
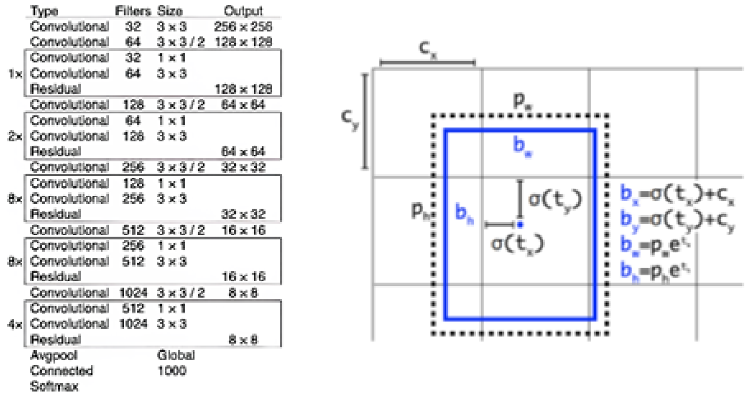


Figure 3. Versions of YOLO algorithm



	Type	Filters	Size	Output
	Convolutional	32	3 × 3	256 × 256
	Convolutional	64	3 × 3 / 2	128 × 128
1×	Convolutional	32	1 × 1	
	Convolutional	64	3 × 3	
	Residual			128 × 128
	Convolutional	128	3 × 3 / 2	64 × 64
2×	Convolutional	64	1 × 1	
	Convolutional	128	3 × 3	
	Residual			64 × 64
	Convolutional	256	3 × 3 / 2	32 × 32
8×	Convolutional	128	1 × 1	
	Convolutional	256	3 × 3	
	Residual			32 × 32
	Convolutional	512	3 × 3 / 2	16 × 16
8×	Convolutional	256	1 × 1	
	Convolutional	512	3 × 3	
	Residual			16 × 16
	Convolutional	1024	3 × 3 / 2	8 × 8
4×	Convolutional	512	1 × 1	
	Convolutional	1024	3 × 3	
	Residual			8 × 8
	Avgpool		Global	
	Connected		1000	
	Softmax			

Figure 4. YOLO bounding box prediction [9]

3.2. YOLOv8 algorithm

YOLOv8, it is a real-time object detecting system that represents YOLO, was used as a previously trained model in this project. Compared to other detection systems [28], YOLOv8 is faster and more accurate, also YOLOv8 can recognize small, forbidden objects with various occlusions while striking an impressive balance between efficiency and detection accuracy. For this reason, YOLOv8 is extremely fast than Fast R-CNN [29]. The architecture of yolov8 strikes a compromise between speed and precision to solve the shortcomings of earlier YOLO iterations [3]. One clear improvement is YOLOv8's scalable and modular architecture. The three main parts of the model are the head, neck, and backbone. YOLOv8's backbone, which consists of CSPDarknet53 and EfficientDet, is accountable for Taking out features from the input image. The fusion of characteristics depends on the neck, which connects the head and backbone. As shown in Figure 5, the head predicts bounding boxes, item classifications, and confidence ratings.

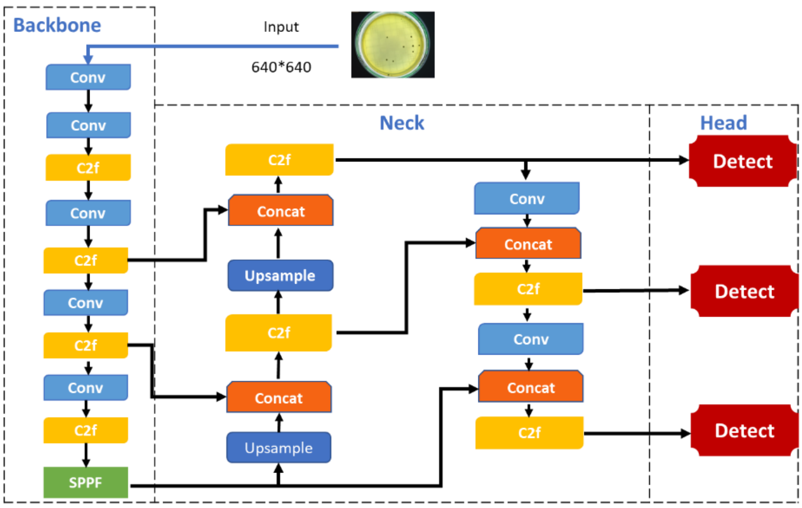


Figure 5. YOLOv8 architecture [3]

3.3. The main criteria of the proposed embedded hardware solution

The objectives to be achieved for the recognition and classification of bacteria, within the framework of this document, must meet the following criteria:

– Provides stable performance in an environment without constraints such as lighting conditions, background inhomogeneity, position and orientation of bacteria, stains and, most importantly, differentiation between a fungus and bacteria.
– Achieves very high recognition rates, demonstrating the reliability of the algorithm's recognition and identification results.
– Since most future applications will be real-time, execution time is also a crucial metric to evaluate the recognition architecture implemented on an embedded system. On a Raspberry Pi 4, for instance, the system is expected to achieve inference latency of approximately 200–300 ms per image, maintain memory usage under 1 GB, and sustain a frame rate of 3–5 FPS depending on image resolution and model complexity [30].
– All the tools chosen are free, such as the Python language and the Raspbian operating system. It is an embedded GNU/Linux operating system compatible with Raspberry Pi microcomputers as presented in the Figure 6.

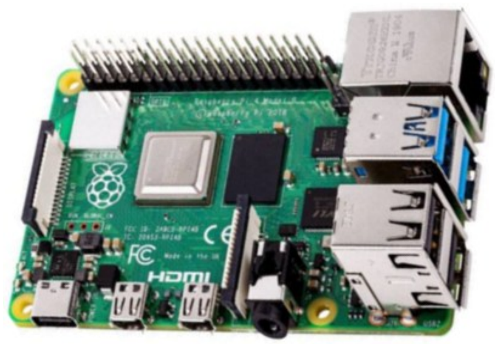


Figure 6. Raspberry Pi 4 version 400

3.4. Methodology adopted

Our study included a number of crucial procedures to identify bacteria and fungus. Since our original dataset was little, we used intensive data augmentation to digitally enlarge it instead of collecting more Petri dish photographs. Using advanced techniques like mosaic, mixup, and copy-paste to increase model robustness, as well as flipping, rotation, scaling, and brightness modifications, augmentation procedures replicated real-world fluctuations in illumination, orientation, and scale.

To produce the labeled dataset, LabelImg was used to annotate two classes: “bac” (bacteria) and “champ” (fungi). To achieve robust evaluation in spite of the small sample size, we employed five-fold cross-validation with train/validation/test splits. This approach produced precise performance estimates for a range of data subsets.

The AdamW optimizer for stable generalization on short datasets, a learning rate of 0.005 with cosine scheduling to enhance convergence, and an image size of 960×960 pixels to capture fine bacterial and fungal features were among the carefully chosen hyperparameters used to train the YOLOv8 model. By balancing accuracy, stability, and computation efficiency, this setup takes into account both realistic deployment aspects and detection performance. Figure 7 illustrates how various combinations improved the accuracy and stability of detection.

3.5. Implementation details

We used a Kaggle notebook environment to train and evaluate the model. The dataset was separated into train, validation, and test subsets to ensure reliable performance estimations, and five-fold cross-validation was employed to lessen bias resulting from the small sample size. The YOLOv8 model was developed using the PyTorch DL platform. For stable optimization, the model was trained for 300 epochs using cosine scheduling with an initial learning rate of 0.005. Based on preliminary experiments and best practices, these hyperparameters were carefully selected. Advanced data augmentation like flipping, rotation, scaling, copy-paste, mosaic, mixup, and brightness adjustment was used to improve model generalization, and the size of the image had been customized to 960×960 pixels to capture fine bacterial and fungal details, all details are shown in Table 1.

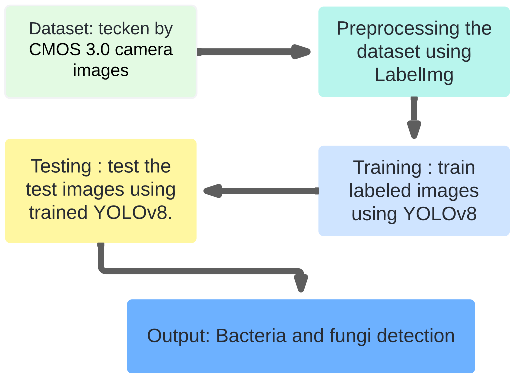


Figure 7. Methodology adopted

4. RESULTS AND DISCUSSION

We successfully applied our method to identify bacteria and fungi in Petri dishes despite working with a comparatively small dataset. With overall scores of precisions 0.864, recall 0.779, and mAP@50 0.859 across all classes, the YOLOv8 model showed excellent performance in recognizing small objects as shown in Table 1. In terms of individual classes, the "champ" class, which had 16 photos with 442 instances, achieved a precision of 0.839, recall of 0.919, and mAP@50 of 0.941, while the "bac" class, which consisted of 23 images with 579 instances, achieved a precision of 0.889, recall of 0.639, and mAP@50 of 0.776. These results are illustrated in Figures 8 and 9, with Figure 10 showing the precision-confidence curve, which demonstrates model precision across all classes as a function of prediction confidence levels.

Table 1. Performance metrics and model details

Learning rate	0.005
Running time	0.149 hours
GPU used	P100
Model size	22.01 Mo
Epochs	300
Precision	86.4%
mAP@50	0. 85

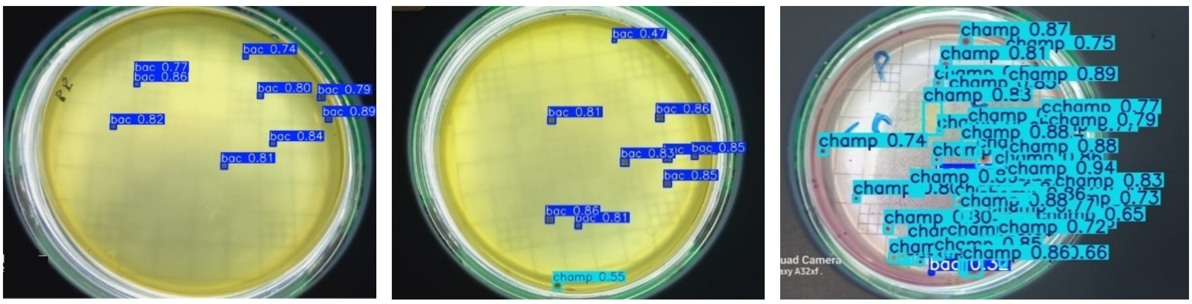


Figure 8. Recognition of bacteria and fungi

Although, Figure 11 confusion matrix demonstrates the suggested model's excellent discriminative power across all three classes. Excellent class separability was confirmed by the champ (fungus) class, which had the best accuracy with 393 correct predictions. With 342 true positives, the bac (bacteria) class also demonstrated strong recognition, proving successful bacterial identification in spite of significant background region confusion. The primary cause of misclassifications between bacteria and background was visual similarity in texture and illumination, but they were still within tolerable bounds. Overall, the confusion matrix confirms the model's capability for real-time bacterial and fungal classification on embedded hardware like the Raspberry Pi and proves its robust and dependable performance.

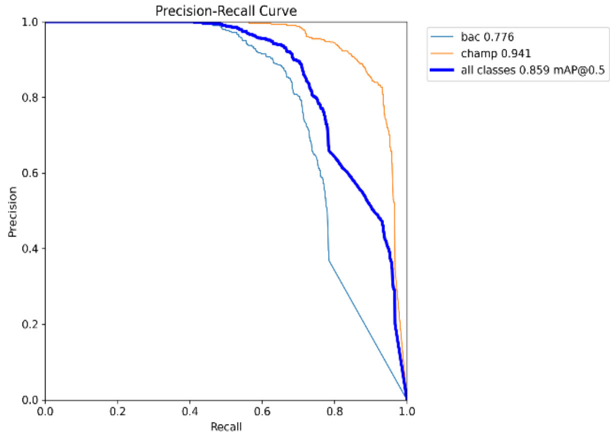


Figure 9. Model performance illustrated by the precision-recall curve

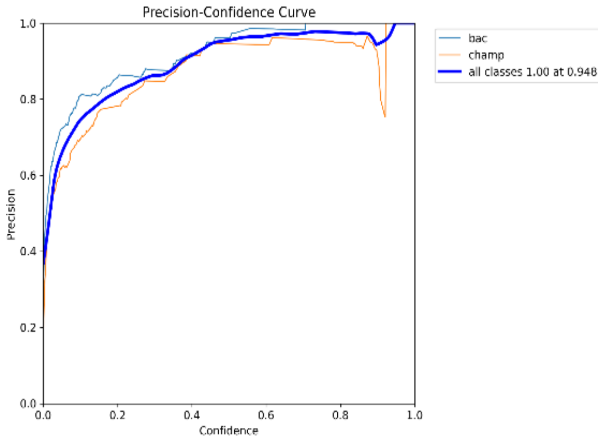


Figure 10. The link between prediction self-assurance and accuracy

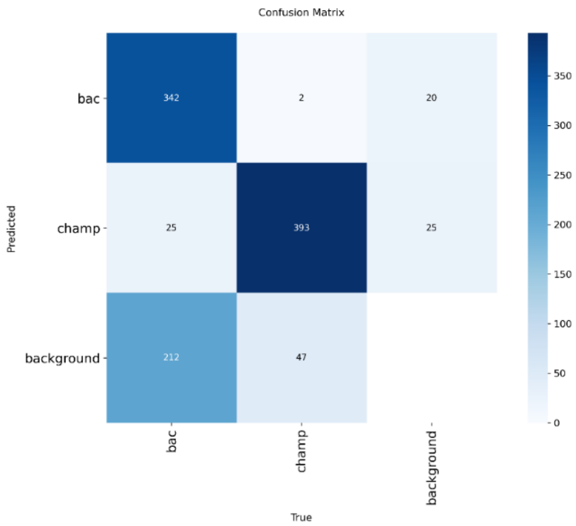


Figure 11. Confusion matrix

Pre-trained model (YOLOv8): although the original YOLO model can detect, count, and recognize several common types of bacteria, it may confuse some unknown species, resulting in lower model accuracy and more manual work carried out by the researchers as presented in Figure 12. YOLOv8 can count and classify several bacteria present in a single image. As you can see in Figure 12, it illustrates the number of bacteria (bac) and fungi (champ) present in each image.

```
!yolo task=detect mode=predict model=/kaggle/working/yolov8/trainingResultsCorrect/BacteriaFun/weights/best.pt source="/kaggle/input/testingmodelimager

Ultralytics YOLOv8.2.61 🚀 Python-3.10.13 torch-2.1.2 CUDA:0 (Tesla T4, 15095MiB)
Model summary (fused): 168 layers, 11,126,358 parameters, 0 gradients, 28.4 GFLOPs

image 1/50 /kaggle/input/testingmodelimagery/testing imagery/A1.jpg: 512x640 9 bacs, 1 champ, 70.8ms
image 2/50 /kaggle/input/testingmodelimagery/testing imagery/A2.jpg: 288x640 17 champs, 70.6ms
image 3/50 /kaggle/input/testingmodelimagery/testing imagery/A4.jpg: 288x640 2 bacs, 49 champs, 10.3ms
image 4/50 /kaggle/input/testingmodelimagery/testing imagery/A5.jpg: 512x640 9 bacs, 13.8ms
image 5/50 /kaggle/input/testingmodelimagery/testing imagery/WhatsApp Image 2024-01-30 � 14.01.24_992e1e5c.jpg: 640x288 5 champs, 68.5ms
image 6/50 /kaggle/input/testingmodelimagery/testing imagery/WhatsApp Image 2024-01-30 � 14.01.28_3d6a64f1.jpg: 640x288 1 bac, 85 champs, 10.3ms
image 7/50 /kaggle/input/testingmodelimagery/testing imagery/WhatsApp Image 2024-01-30 � 14.01.31_aa8da2b0.jpg: 640x288 65 champs, 10.3ms
image 8/50 /kaggle/input/testingmodelimagery/testing imagery/WhatsApp Image 2024-01-30 � 14.01.33_6edbaf59.jpg: 288x640 142 bacs, 10.9ms
image 9/50 /kaggle/input/testingmodelimagery/testing imagery/WhatsApp Image 2024-01-30 � 14.01.36_7e21e8d5.jpg: 288x640 146 bacs, 2 champs, 10.3ms
image 10/50 /kaggle/input/testingmodelimagery/testing imagery/WhatsApp Image 2024-01-30 � 14.01.50_dadd36f0.jpg: 288x640 5 bacs, 10.3ms
image 11/50 /kaggle/input/testingmodelimagery/testing imagery/WhatsApp Image 2024-01-30 � 14.02.11_36018d1e.jpg: 288x640 4 bacs, 10.3ms
image 12/50 /kaggle/input/testingmodelimagery/testing imagery/WhatsApp Image 2024-01-30 � 14.02.16_a47d8323.jpg: 288x640 3 bacs, 91 champs, 10.3ms
image 13/50 /kaggle/input/testingmodelimagery/testing imagery/WhatsApp Image 2024-01-30 � 14.02.17_58d69107.jpg: 288x640 17 bacs, 10.3ms
image 14/50 /kaggle/input/testingmodelimagery/testing imagery/WhatsApp Image 2024-01-30 � 14.02.17_b1ee5b12.jpg: 288x640 62 bacs, 10.3ms
image 15/50 /kaggle/input/testingmodelimagery/testing imagery/WhatsApp Image 2024-01-30 � 14.02.18_745f046e.jpg: 288x640 5 bacs, 77 champs, 10.3ms
image 16/50 /kaggle/input/testingmodelimagery/testing imagery/WhatsApp Image 2024-01-30 � 14.02.19_88bd7627.jpg: 288x640 17 bacs, 10.3ms
image 17/50 /kaggle/input/testingmodelimagery/testing imagery/WhatsApp Image 2024-01-30 � 14.02.19_98a9c4ad.jpg: 288x640 5 bacs, 10 champs, 10.3ms
image 18/50 /kaggle/input/testingmodelimagery/testing imagery/WhatsApp Image 2024-01-30 � 14.02.19_f44852e7.jpg: 288x640 3 bacs, 50 champs, 10.3ms
image 19/50 /kaggle/input/testingmodelimagery/testing imagery/WhatsApp Image 2024-01-30 � 14.02.20_478b13d1.jpg: 288x640 3 bacs, 129 champs, 10.3ms
image 20/50 /kaggle/input/testingmodelimagery/testing imagery/WhatsApp Image 2024-01-30 � 14.02.21_64c17671.jpg: 640x288 10 champs, 11.0ms
image 21/50 /kaggle/input/testingmodelimagery/testing imagery/WhatsApp Image 2024-01-30 � 14.02.21_edd2d4b4.jpg: 288x640 25 bacs, 62 champs, 11.2ms
image 22/50 /kaggle/input/testingmodelimagery/testing imagery/WhatsApp Image 2024-01-30 � 14.02.22_89738b60.jpg: 288x640 2 bacs, 5 champs, 10.3ms
image 23/50 /kaggle/input/testingmodelimagery/testing imagery/WhatsApp Image 2024-01-30 � 14.02.22_aace6a45.jpg: 288x640 64 bacs, 10.3ms
image 24/50 /kaggle/input/testingmodelimagery/testing imagery/WhatsApp Image 2024-01-30 � 14.02.23_4b3e38a9.jpg: 288x640 7 bacs, 14 champs, 10.3ms
image 25/50 /kaggle/input/testingmodelimagery/testing imagery/WhatsApp Image 2024-01-30 � 14.02.24_d9160335.jpg: 640x288 4 bacs, 60 champs, 11.0ms
```

Figure 12. Counting several bacteria

Refined model (result): to evaluate the model, photos of Petri dishes taken by doctoral students in the microbiology laboratory at CNRST were used as a test set to which the trained model is applied. The results illustrate that the DL model was able to detect, recognize, and localize multiple bacteria and differentiate between bacteria (small sizes with the same colors) and fungi (larger sizes and different colors) via CMOS 3.0 camera images with great precision. Each identified object will be surrounded by a bounding box and labeled with its species and order number [31]. Practically speaking, this method may be immediately used in microbiological labs to help researchers automate morphological characterization, colony counting, and bacterial and fungal distinction, greatly minimizing manual labor and human error. This kind of connection could improve the repeatability of microbiological tests and expedite laboratory procedures.

Nevertheless, several restrictions were noted. The proposed model may not be as generalizable to larger microbiological contexts because it was trained and validated on a small dataset with little species variety. Furthermore, the moderate recall rate suggests that some bacterial colonies are still overlooked during detection. Future research will concentrate on examining advanced architectures like vision transformers (ViTs) for more precise detection of small and visually similar microbial objects, applying transfer learning from large-scale biomedical image datasets to improve feature generalization, and broadening the dataset to include a greater variety of microbial species in order to get around these limitations.

5. CONCLUSION

This project aims to train an algorithm capable of processing and labeling data from Petrie dishes with the number, type, and location of bacteria and fungi. Captured huge and laborious images in an automatic way that researchers were working by applying a YOLOv8-based object detection. Compared to other attempts at automatic bacteria recognition, this project strives to further facilitate the process and reduce the human effort required by providing surrounding bounding boxes, performing real-time recognition, and adding bacteria that can be automatically identified. This research addresses major data issues by creating a microbiological recognition model. We used the YOLOv8 object detection technique as the pre-trained model using transfer learning. Even with a rather small image database, we were able to identify fungi and bacteria in Petri dishes. The model's excellent accuracy in recognizing and localizing bacteria was achieved with additional refinement using better data that was first annotated by microbiology professionals. This illustrates the success of YOLO algorithms in these applications and helps alleviate the workload from human for tasks such as recognition and counting. In order to enhance the performance and accuracy in detection, especially for small objects, the future work will focus on designing powerful algorithms which applying high performance CPU and extending the plate types as well as variety of bacteria collection.

ACKNOWLEDGMENTS

The authors express their thanks to Mohammed V University, Faculty of Sciences and CNRST Laboratory for further guidance and providing necessary information on this research topic doctoral about the bacterial and fungal identification in Petri dishes. The authors also thank all those who helped with technical assistance or advice.

FUNDING INFORMATION

The authors are grateful to the institution for support. No specific grant, funding organization or financial support was received for this study.

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
Tarik Bouganssa	✓	✓	✓	✓	✓	✓		✓	✓	✓	✓			✓
Maryem Ait Moulay	✓	✓		✓	✓	✓	✓		✓	✓	✓	✓		✓
Samar Aarabi	✓			✓						✓	✓			
Abdelali Lasfar	✓			✓	✓					✓	✓	✓	✓	
Abdellatif El Afia				✓	✓					✓	✓	✓	✓	✓

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

The authors affirm that there are no known financial, personal, or professional ties that could have affected the findings or interpretations offered in this study in order to maintain objectivity and transparency.

DATA AVAILABILITY

The data that support the findings of this study are available from the corresponding author, upon reasonable request.

REFERENCES

[1] M. M. Ansari, N. Bisht, T. Singh, and P. S. Chauhan, "Symphony of survival: insights into cross-talk mechanisms in plants, bacteria, and fungi for strengthening plant immune responses," *Microbiological Research*, vol. 285, Aug. 2024, doi: 10.1016/j.micres.2024.127762.

[2] M. Willi *et al.*, "Identifying animal species in camera trap images using deep learning and citizen science," *Methods in Ecology and Evolution*, vol. 10, no. 1, pp. 80–91, Jan. 2019, doi: 10.1111/2041-210X.13099.

[3] M. A. Moulay, A. Salbi, I. Bouganssa, M.-S. Masmoudi, and A. Lasfar, "An innovative approach for detecting buildings and construction anomalies in zenata city," *IAES International Journal of Artificial Intelligence*, vol. 13, no. 3, pp. 2703-2712, Sep. 2024, doi: 10.11591/ijai.v13.i3.pp2703-2712.

[4] C. S. Costa *et al.*, "Deep learning applied in fish reproduction for counting larvae in images captured by smartphone," *Aquacultural Engineering*, vol. 97, May 2022, doi: 10.1016/j.aquaeng.2022.102225.

[5] I. Ahmed, M. Ahmad, and G. Jeon, "Integrating digital twins and deep learning for medical image analysis in the era of covid-19," *Virtual Reality & Intelligent Hardware*, vol. 4, no. 4, pp. 292–305, Aug. 2022, doi: 10.1016/j.vrih.2022.03.002.

[6] X. Chen, H. Zheng, H. Tang, and F. Li, "Multi-scale perceptual yolo for automatic detection of clue cells and trichomonas in fluorescence microscopic images," *Computers in Biology and Medicine*, vol. 175, Jun. 2024, doi: 10.1016/j.compbiomed.2024.108500.

[7] J. Li, W. Chen, Y. Zhu, K. Xuan, H. Li, and N. Zeng, "Intelligent detection and behavior tracking under ammonia nitrogen stress," *Neurocomputing*, vol. 559, Nov. 2023, doi: 10.1016/j.neucom.2023.126809.

[8] F. Jubayer *et al.*, "Detection of mold on the food surface using YOLOv5," *Current Research in Food Science*, vol. 4, pp. 724–728, 2021, doi: 10.1016/j.crfs.2021.10.003.
[9] S. Schneider, G. W. Taylor, and S. Kremer, "Deep learning object detection methods for ecological camera trap data," in *2018 15th Conference on Computer and Robot Vision (CRV)*, IEEE, May 2018, pp. 321–328, doi: 10.1109/CRV.2018.00052.
[10] S. Aarabi, O. Chauiyakh, T. Bouganssa, E. E. Fahime, and A. Et-tahir, "Bioaccumulation of heavy metals in five species of fish obtained from the estuary of rabat, morocco," *Tropical Journal of Natural Product Research*, vol. 8, no. 4, May 2024, doi: 10.26538/tjnpr/v8i4.3.
[11] Q. Yao, X. Zheng, G. Zhou, and J. Zhang, "SGR-yolo: a method for detecting seed germination rate in wild rice," *Frontiers in Plant Science*, vol. 14, Jan. 2024, doi: 10.3389/fpls.2023.1305081.
[12] D. Doherty *et al.*, "Assessing the viability of 3d-printed poly (lactic acid) petri dishes: a sustainable alternative for laboratory use," *Sustainable Materials and Technologies*, vol. 40, Jul. 2024, doi: 10.1016/j.susmat.2024.e00899.
[13] Q. Wu *et al.*, "The screening and identification of six commonly used antibiotics in swine kidney by a microbiological inhibition method," *Microchemical Journal*, vol. 161, Feb. 2021, doi: 10.1016/j.microc.2020.105796.
[14] E. A. Landis, E. Fogarty, J. C. Edwards, O. Popa, A. M. Eren, and B. E. Wolfe, "Microbial diversity and interaction specificity in kombucha tea fermentations," *mSystems*, vol. 7, no. 3, Jun. 2022, doi: 10.1128/msystems.00157-22.
[15] N. R. Mannuru *et al.*, "Artificial intelligence in developing countries: the impact of generative artificial intelligence (AI) technologies for development," *Information Development*, vol. 41, no. 3, pp. 1036–1054, 2025, doi: 10.1177/02666669231200628.
[16] R. Bunod, E. Augstburger, E. Brasnu, A. Labbe, and C. Baudouin, "Intelligence artificielle et glaucome : une revue de la littérature," *Journal Français d'Ophtalmologie*, vol. 45, no. 2, pp. 216–232, Feb. 2022, doi: 10.1016/j.jfo.2021.11.002.
[17] J. Kim *et al.*, "A novel deep learning–based artificial intelligence system for interpreting urolithiasis in computed tomography," *European Urology Focus*, vol. 10, no. 6, pp. 1049–1054, Dec. 2024, doi: 10.1016/j.euf.2024.07.003.
[18] J. Terven, D. M. C. Esparza, and J. A. R. 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.
[19] K. Alomar, H. I. Aysel, and X. Cai, "CNNs, rnns and transformers in human action recognition: a survey and a hybrid model," *Artificial Intelligence Review*, vol. 58, no. 12, Oct. 2025, doi: 10.1007/s10462-025-11388-3.
[20] I. Bouganssa, H. Benradi, A. Salbi, M. Sbihi, and A. Lasfar, "Embedded fpga hardware implementation of a preprocessing algorithm for surveillance images," *Journal of Theoretical and Applied Information Technology*, vol. 101, no. 19, pp. 6001–6012, 2023.
[21] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: unified, real-time object detection," *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA,* 2016, pp. 779-788, doi: 10.1109/CVPR.2016.91.
[22] C. M. Badgujar, A. Poulose, and H. Gan, "Agricultural object detection with you only look once (yolo) algorithm: a bibliometric and systematic literature review," *Computers and Electronics in Agriculture*, vol. 223, Aug. 2024, doi: 10.1016/j.compag.2024.109090.
[23] Z. Cai, Y. Zhang, J. Li, J. Zhang, and X. Li, "Synchronous detection of internal and external defects of citrus by structured-illumination reflectance imaging coupling with improved YOLO v7," *Postharvest Biology and Technology*, vol. 227, Sep. 2025, doi: 10.1016/j.postharvbio.2025.113576.
[24] Sandhya and A. Kashyap, "A novel method for real-time object-based copy-move tampering localization in videos using fine-tuned YOLO v8," *Forensic Science International: Digital Investigation*, vol. 48, Mar. 2024, doi: 10.1016/j.fsidi.2023.301663.
[25] P. Jiang, D. Ergu, F. Liu, Y. Cai, and B. Ma, "A review of YOLO algorithm developments," *Procedia Computer Science*, vol. 199, pp. 1066–1073, 2022, doi: 10.1016/j.procs.2022.01.135.
[26] H. Chen and J. Tao, "Utilizing improved yolov8 based on SPD-BRSA-AFPN for ultrasonic phased array non-destructive testing," *Ultrasonics*, vol. 142, Aug. 2024, doi: 10.1016/j.ultras.2024.107382.
[27] C. Chen, H. Lee, and M. Chen, "Steel surface defect detection method based on improved YOLOv9," *Scientific Reports*, vol. 15, no. 1, 2025, doi: 10.1038/s41598-025-10647-1.
[28] S. Sun, B. Mo, J. Xu, D. Li, J. Zhao, and S. Han, "Multi-YOLOv8: an infrared moving small object detection model based on YOLOv8 for air vehicle," *Neurocomputing*, vol. 588, Jul. 2024, doi: 10.1016/j.neucom.2024.127685.
[29] A. S. M. M. Hasan, D. Diepeveen, H. Laga, M. G. K. Jones, and F. Sohel, "Object-level benchmark for deep learning-based detection and classification of weed species," *Crop Protection*, vol. 177, Mar. 2024, doi: 10.1016/j.cropro.2023.106561.
[30] H. Benradi, I. Bouganssa, A. Chater, and A. Lasfar, "Discriminative approach lung diseases and covid-19 from chest x-ray images using convolutional neural networks: a promising approach for accurate diagnosis," *International Journal of Online and Biomedical Engineering*, vol. 19, no. 14, pp. 131–141, 2023, doi: 10.3991/ijoe.v19i14.42725.
[31] J. Ding, J. Zhang, Z. Zhan, X. Tang, and X. Wang, "A precision efficient method for collapsed building detection in post-earthquake uav images based on the improved NMS algorithm and faster R-CNN," *Remote Sensing*, vol. 14, no. 3, Jan. 2022, doi: 10.3390/rs14030663.

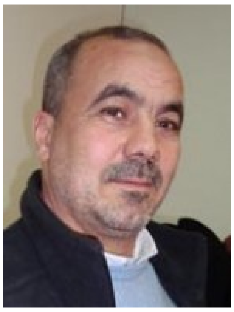
BIOGRAPHIES OF AUTHORS

Tarik Bouganssa received his Ph.D. student in Information Technology and Engineering Sciences option: Applied Mathematics and Artificial Intelligence, Laboratory: Smart Systems Laboratory (SSL) at the National School of Computer Science and Systems Analysis (ENSIAS), Mohammed V University of Rabat. Master of Research in Mathematics and Computer Science (2021) option: data science and big data at the National School of Computer Science and Systems Analysis (ENSIAS), Mohammed V University of Rabat. Bachelor of Fundamental Studies (2009) in Mathematical Sciences and Computer Science (SMI) at the Faculty of Sciences of Rabat FSR, Mohammed V University Rabat-Agdal. He can be contacted at email: tarik.bouganssa@gmail.com.

Maryem Ait Moulay is a Ph.D. student at the Laboratory Systems Analysis, Information Processing, and Industrial Management, Mohammadia School of Engineering (EMI), Mohammed V University of Rabat, Morocco. Passionate about artificial intelligence. Her focus is on exploring innovative applications in artificial intelligence, computer vision, and information processing. She can be contacted at email: maryem.aitmoulay29@gmail.com.

Samar Aarabi received her Ph.D. in Water and Environmental Engineering (2024) within the research team Materials, Energy and Acoustics Team (MEAT) at EST-Salé CEDOC of the Mohammadia School of Engineers, Mohammed V University in Rabat. Her thesis subject: the degradation of fish quality and the impact of trace elements on human health. Specialized master (2016) in Quality Assurance of Medicines at the Faculty of Medicine and Pharmacy of Rabat, Mohammed V University. Fundamental license (2014) in Life Sciences, biodiversity and environment option at the Faculty of Sciences, Mohammed V University of Rabat. She can be contacted at email: samar.aarabi1@gmail.com.

Abdelali Lasfar is a professor of Higher Education at Mohammed V Agdal University, Salé Higher School of Technology, Morocco. His research focuses on compression methods, indexing by image content and image at the LASTIMI Laboratory. He can be contacted at email: abdelali.lasfar@est.um5.ac.ma.



Abdellatif El Afia is a full professor in the Department of Computer Science and Decision Support, holding a Ph.D. in Operations Research from the Université de Sherbrooke (1999). He currently serves as the Coordinator of the Artificial Intelligence Engineering (2IA) program. His research interests are primarily centered on Mathematical Programming (both deterministic and stochastic), various facets of machine learning (including supervised, statistical, and reinforcement learning), the tuning of metaheuristics, and the performance analysis of both learning and optimization algorithms. He can be contacted at email: abdelatif.elafia@um5.ac.ma.