Deep learning architectures for location and identification in storage systems

Anny Astrid Espitia Cubillos[1], Robinson Jiménez-Moreno[2], Esperanza Rodríguez Carmona[1]
[1]Industrial Engineering Program, Faculty of Engineering, University Militar Nueva Granada, Bogotá, Colombia
[2]Mechatronic Engineering Program, Faculty of Engineering, University Militar Nueva Granada, Bogotá, Colombia

Article Info

Article history:

Received Apr 2, 2024
Revised Jul 27, 2024
Accepted Aug 30, 2024

Keywords:

Accurancy
Convolutional networks
Deep learning
Products identification
Products location
Storage systems

ABSTRACT

This document exposes the application of two deep learning models based on ResNet-18 architectures, intended for the location and identification of products in storage areas. One model obeys a tree structure and the other a structure under an ouroboron cycle. The performance of both models is evaluated using the metrics of training time, processing time and level of learning precision, which allows recommendations to be made regarding which one should be used for order preparation purposes, based on multilevel feature extraction. The total training time of the first model is 34.65 minutes and the second 40.43 minutes. The analysis of results allowed the detection parameters to be adjusted, finally with the refined models, through confusion matrices, precision results greater than 90% and processing times are obtained, which for model 1 is 6.8565 seconds and for model 2 is 4.884 seconds. For practical purposes, training times are not relevant, as are the precision and processing times for selecting the most convenient model according to the end user's objectives.

Corresponding Author:

Anny Astrid Espitia Cubillos
Industrial Engineering Program, Faculty of Engineering, Universidad Militar Nueva Granada
Carrera 11 #101-80, Bogotá, Colombia
Email: anny.espitia@unimilitar.edu.co

1. INTRODUCTION

Currently, concepts and implementation of industry 4.0 are a reality in production at various levels [1], where integration of data analysis, sensors and artificial intelligence go hand in hand with schemes that revolutionize industrial processes, such as inclusion of robots [2]. Being able to measure and act on outputs of a production line are tasks inherent to development of industry 4.0 and internet of things (IoT)-based systems [3]. All of this allows automation in production lines that translates into lower costs and times, under the smart factory concept [4]. Storage of products and their location within collection site are tasks that can be automated given high volume of demand and standardization of the process. In the case of an automated system that allows this task to be achieved, it must identify the storage location and discriminate each product among the possible ones that are stored, this as preliminary activities for order preparation.

In artificial intelligence, the use of deep learning algorithms applied to manufacturing processes in industry stands out [5]. Where the integration of deep learning with robotic automation systems allows automating factory processes such as product inventories [6]. These algorithms today demarcate the implementation of Industry 4.0 schemes, based on techniques for multidimensional pattern recognition [7], where convolutional neural networks (CNN) are presented as the predominant technique. A convolutional network is a layer-based learning architecture, where the simplest structure is the union of convolution, linear rectification, and pooling operations for resizing. These are two-dimensional order operations carried out through matrix-type operations [8], highly efficient for detection of objects in images.

Object detection corresponds to locating all positions of objects to be verified in an entry that is delimited in boxes and labeled according to category to which it belongs [9]. A variety of applications employ deep learning for Industry 4.0, such as computer vision-based supermarket merchandise management using CNN, YOLOv3, and Keras [10]. Given advancement of deep learning models and artificial intelligence technologies, real-time object detection methods are an area of study to improve their effectiveness and potential for practical applications, including objects with geometric diversity and variety of optical properties [11]. This is reflected in management of components in supply chains in automotive industry and manufacturing industry, for example, to detect assembly parts [12]. Also, in identification of objects with deformed parts using geometric restriction and penalty [13] and in construction industry by categorizing objects by worker, material, machine and design [14].

Object detection performance has improved through use of latest generation models based on both single-stage and two-stage deep learning [12], [15]. Although facilities of systems based on industry 4.0, today they can help prevent and/or correct risk situations in workers [16], even musculoskeletal risks from repetitive tasks remain high [17]. One of factors of this type of risk is task of storing goods, for example stacking boxes, a task that can be automated. Therefore, the objective of this article is to design and evaluate two pattern recognition methods for identifying boxes in industrial environments, which can be manipulated by robotic agents in a second phase. For this, convolutional networks are used with detection of region-based convolutional neural networks (RCNN) regions [18], based on multilevel detection of patterns, at first level the detection of boxes and at second level the detection of particular characteristics that allow them to be classified.

This article is divided into four sections, the first with the introduction to the state of the art and proposed development. The second section presents the two proposed methods. The third section presents the analysis and discussion of results. Finally the conclusions reached are presented.

2. METHOD

In order to identify and locate a product (box) within a storage area such as a shelf, starting point is to establish a test scenario in which region-based deep learning networks will be used, steps to follow are shown graphically in Figure 1, for this, the most convenient predefined architecture is selected according to characteristics to be discriminated. Subsequently, solutions are designed that respond to what is required: identify each box and its type. To train network, database is built, network architecture is determined, training parameters are selected as the most appropriate optimizer, and validation tests are performed to identify and document errors. Finally, the model is refined and the metrics of processing time and level of accuracy in final recognition are calculated to give a recommendation in this regard.

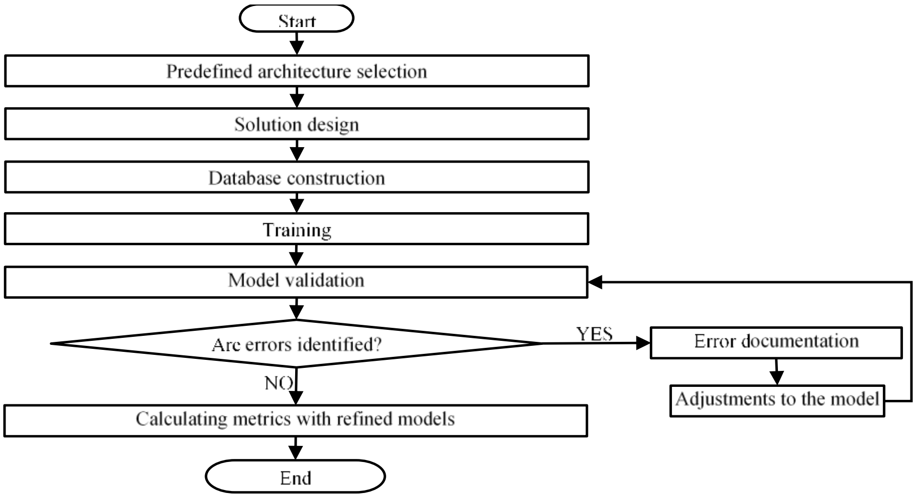


Figure 1. Methodology flowchart

3. RESULTS AND DISCUSSION

Considering robust developments in predefined CNN-based architectures, such as ResNet-18 [18], [19], ResNet-50 [20], ResNet-101 [21], and similar, it is decided to use a transfer model of learning. For this case, discrimination of storage boxes labeled for classification and located on a shelf is required. For its evaluation, a simulation of three-level storage shelves and boxes labeled with the letters A, B, C will be used as shown in Figure 2. The boxes will be of different types that allow, if necessary, to have different products internally or same product in different quantities, using various sizes of boxes. Taking into account the low number of characteristics to be discriminated, it is decided to use a transfer learning model to centralize the design in the most favorable global algorithmic structure and not in the network architecture. For this case, training based on the ResNet-18 architecture [18], is chosen, a shallow classification and localization architecture that reduces the computational cost.

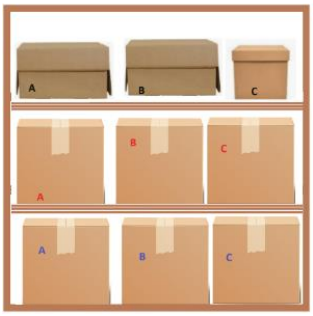


Figure 2. Storage shelf example

Since the aim is to locate each box on the shelf and subsequently identify the type of box, two particular solutions are proposed. First a tree structure where two ResNet-18 architectures are trained [22], the top node that will detect boxes and after that the network of the bottom node that will detect the type of box. Second solution uses same network with joint training of boxes and box type label, where network output, with box identification, is fed back for later identification of box type, and again until all types of boxes present are labeled (emulating an ouroboro cycle). Figures 3(a) and 3(b) respectively illustrates flowchart of proposed solutions.

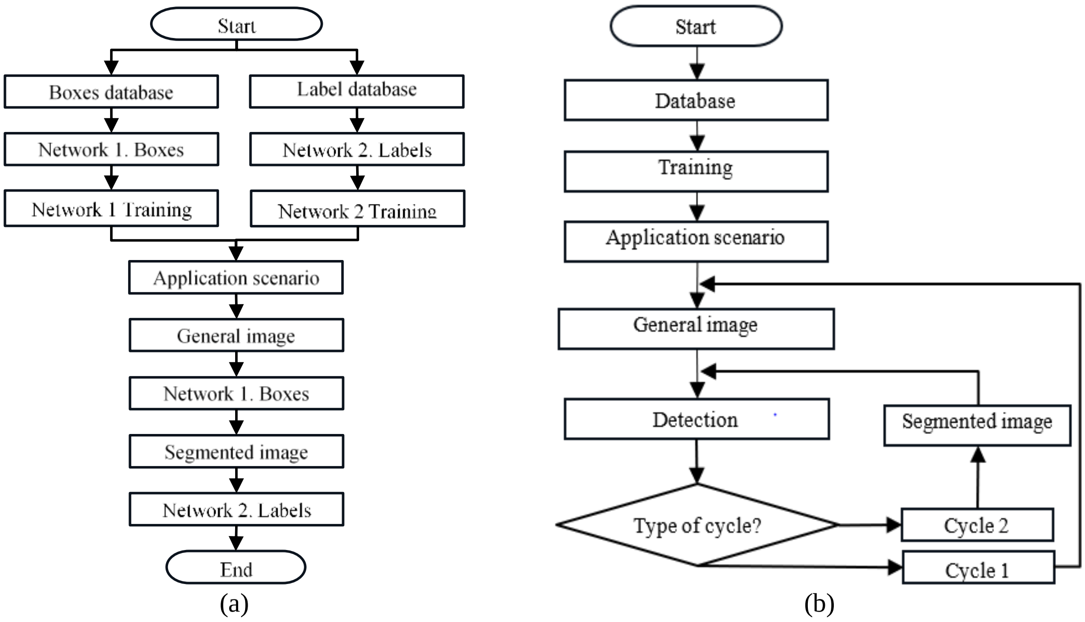


Figure 3. Flowchart of (a) tree structure and (b) ouroboro structure

For training of each solution, a database is used that is built based on desired objective. Figure 4 illustrates part of database of box storage shelves, where boxes of different types and sizes are used. Each image has a resolution of 1,000×1,000 pixels and 100 images of this type are used. Figure 5 presents a portion of database of boxes with discrimination labels, in which case boxes of type A, B, or C will be recognized without discriminating their size or shape. Each image has a resolution of 224×224 pixels (ResNet-18 input) and 100 images of this type are used. Figure 6 shows network architecture used, where it is important to highlight that ResNet-18 has 18 layers with hyperparameters defined as the size of filters and convolution and pooling operations [8], [23]. Therefore, in concept of transfer learning, only the type of network training parameters is determined, as illustrated in Table 1.

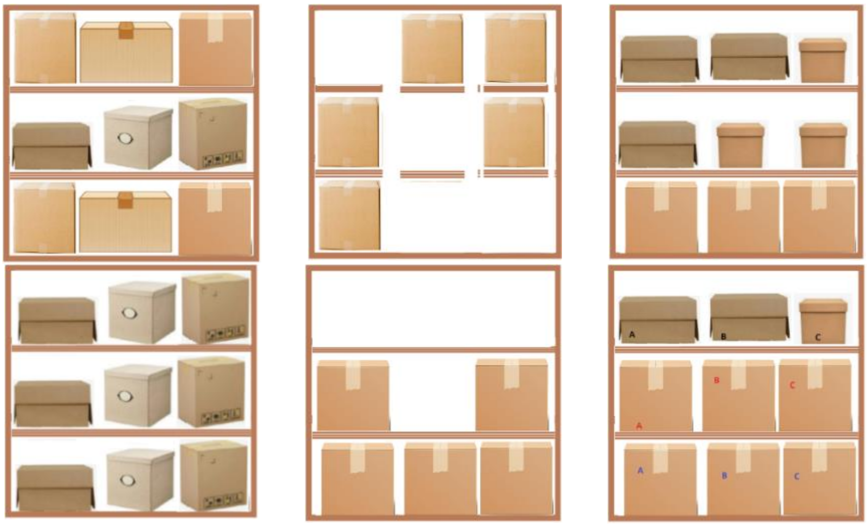


Figure 4. Database example of boxes on storage shelf

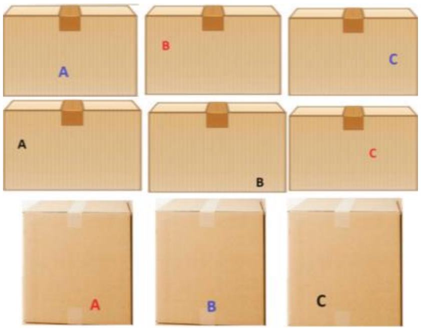


Figure 5. Discrimination labeled boxes database

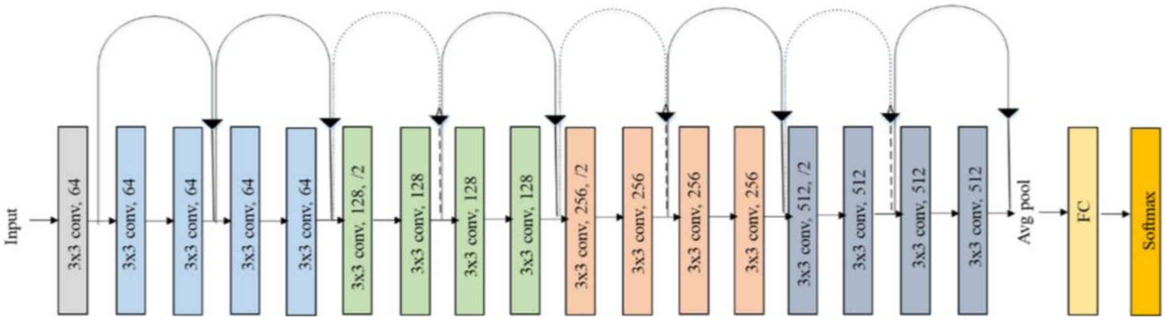
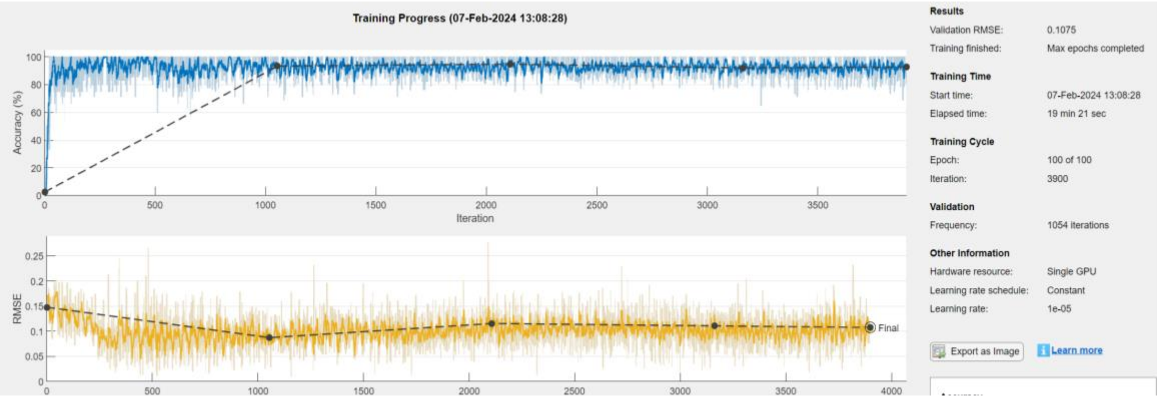


Figure 6. ResNet-18 architecture

In this case, images are resized to 224×224 pixels because it is input size of the network (224×224×3). Resolution reduction generates noise in training that makes Adam optimizer present better performance than one based on stochastic gradient descent (SGD) [24]. Figures 7 to 9 show results of training networks on a computer with an RTX-4070 GPU with 8GB of memory.

Table 1. Training parameters

Parameter	Setting value
Training Options	adam
Mini Batch Size	1
Initial Learn Rate	10e-6
Max Epochs	100
Validation Data	Preprocessed Training Data
Validation Frequency	1,054



Results
Validation RMSE: 0.1075
Training finished: Max epochs completed

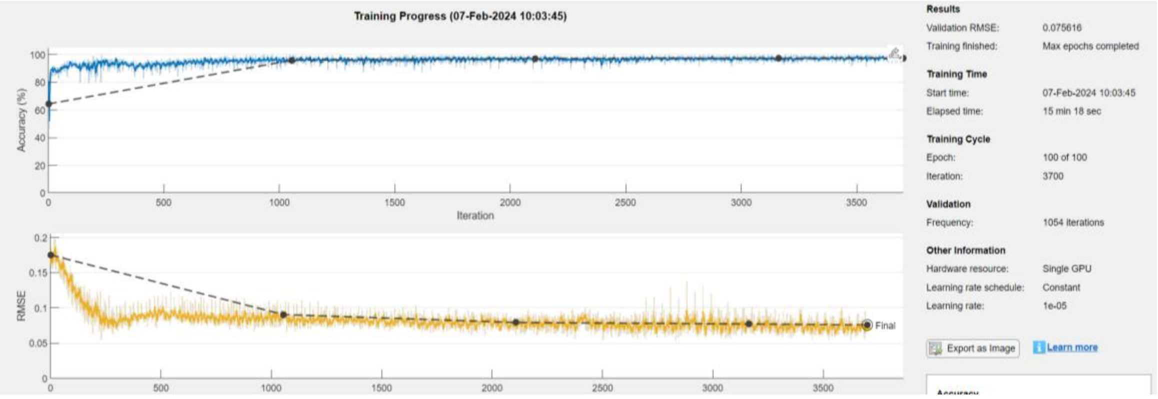
Training Time
Start time: 07-Feb-2024 13:08:28
Elapsed time: 19 min 21 sec

Training Cycle
Epoch: 100 of 100
Iteration: 3900

Validation
Frequency: 1054 iterations

Other Information
Hardware resource: Single GPU
Learning rate schedule: Constant
Learning rate: 1e-05

Figure 7. First tree network (training boxes)



Results
Validation RMSE: 0.075616
Training finished: Max epochs completed

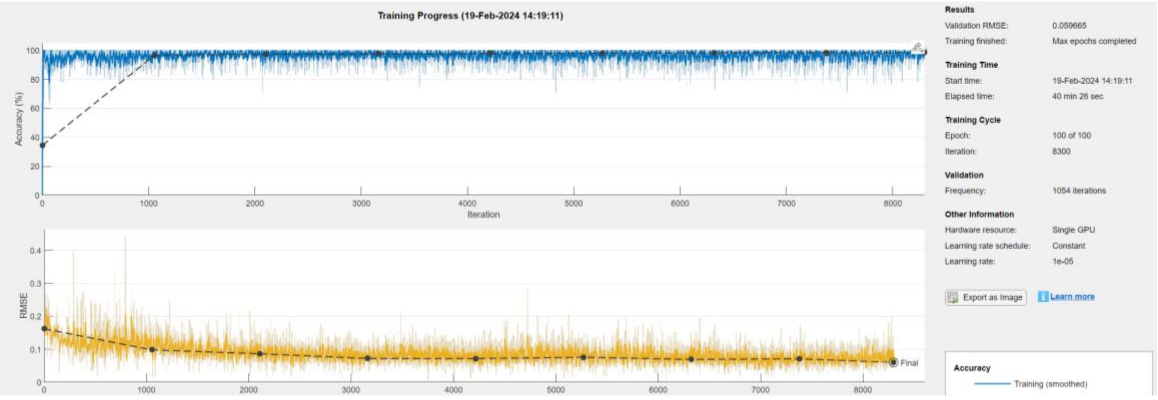
Training Time
Start time: 07-Feb-2024 10:03:45
Elapsed time: 15 min 18 sec

Training Cycle
Epoch: 100 of 100
Iteration: 3700

Validation
Frequency: 1054 iterations

Other Information
Hardware resource: Single GPU
Learning rate schedule: Constant
Learning rate: 1e-05

Figure 8. Second tree network (training labels)



Results
Validation RMSE: 0.059665
Training finished: Max epochs completed

Training Time
Start time: 19-Feb-2024 14:19:11
Elapsed time: 40 min 26 sec

Training Cycle
Epoch: 100 of 100
Iteration: 8300

Validation
Frequency: 1054 iterations

Other Information
Hardware resource: Single GPU
Learning rate schedule: Constant
Learning rate: 1e-05

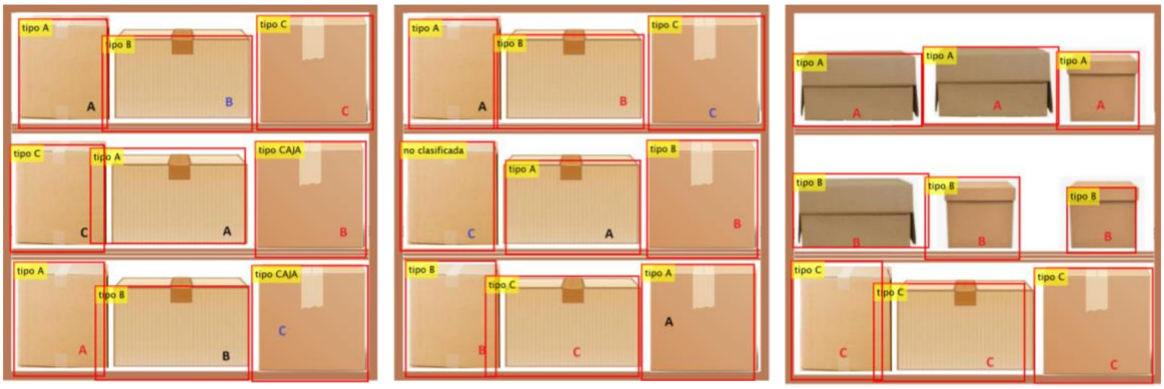
Figure 9. Ouroboro model network training (training boxes and labels)

A comparison of training results is presented in Table 2, it is observed that sum of training time of the tree model networks is 34.65 minutes, each with its respective database (boxes and labels). For the ouroboro model network, an integrated database of boxes and labels is used, training took 40.43 minutes. For both cases, precision levels greater than 95% are obtained in learning. Root mean square error (RMSE) is an indicator to evaluate quality of predictions for a particular data set and not between data sets, since it depends on scale used, it shows to what extent predictions vary from true values using Euclidean distance, therefore, values close to zero are desired, in this case, it is observed that training is more accurate for the ouroboro model when compared with average value of the tree model. Almalaq and Edwards [25] compares accuracy results using RMSE for various applications using CNN. Table 2 also shows that in total 7,600 iterations were carried out in the tree model network while the ouroboro model network required 8,300 for its training.

Table 2. Training results

Results	Tree model			Ouroboro model
	Boxes	Labels	Total/Average	
Validation RMSE	0.1075	0.075616	0.091558	0.059665
Training time	19,35 minutes	15,3 minutes	34,65 minutes	40,43 minutes
Iterations	3,900	3,700	7,600	8,300

After training, validation tests are carried out with various results such as those presented in Figure 10. Errors are observed in the classification as shown Figure 10(a) where two of the nine boxes are not labeled correctly, this error is derived from model used, for the ouroboro case, as will be explained later. Another error is evident in non-detection of label as shown Figure 10(b). Finally, Figure 10(c) exposes one of several cases of correct classifications.



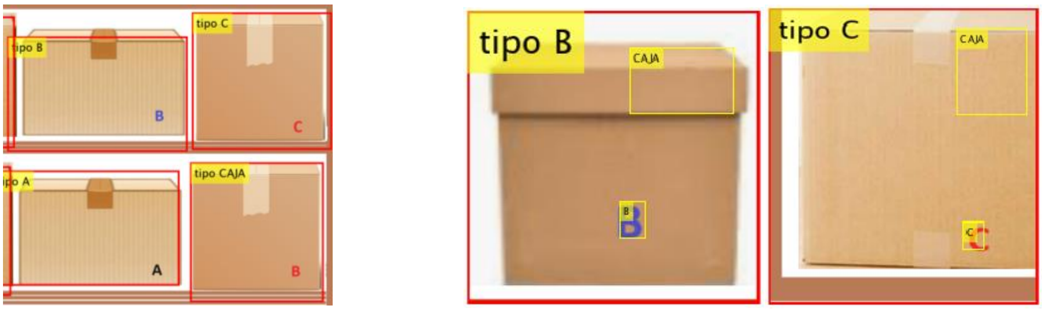
(a) (b) (c)

Figure 10. Ouroboro structure results (a) classification with errors, (b) unclassified, and (c) correct classification

Figure 11 shows cause of mislabeling, validating with several examples, is evidently derived from the same ouroboro model. That is, when re-entering box detection image from first time to network, as model is trained with box database, it also seeks to identify these. Figure 11(a) shows a box type classification error, while Figure 11(b) shows a box identification error. Size of box it uses for object detection, biased to the minimum of label, it sees parts of same box as a new box, this error when identified is eliminated with software filtering in identification of label type, deleting everything that is marked as a box and establishing cycles in detection, one general for the box and next for the label in the ouroboro model.

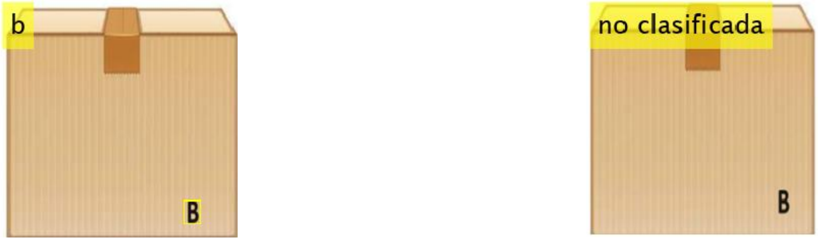
Error of non-detection of label occurs because training box type is taken at 224×224 pixels in database. When each box is detected and extracted from original 1,000×1,000-pixel image to preserve resolution, it has sizes greater than 224×224 pixels, so it must be resized to network input size. Rectangular boxes, when resized, modify shape and width of label, making it not legible in some cases, which was solved by entering these cases into database. Figure 12 illustrates the way in which the error was evident. In this case, an example was taken from database to validate correct classification. Figure 12(a) and later it is resized slightly smaller and when evaluated through network, it does not distinguish label and therefore does not classify box as shown Figure 12(b).

Figure 13 relates detection for case of the tree-type model, there are two scenarios, the first that shows an example of correct operation in Figure 13(a) and the second where the absence of classification is evident as shown in Figure 13(b), this error was solved in same way as error of non-detection of the ouroboro model label since it was due to same cause. Once adjustments have been made to detected errors, refined models are evaluated using confusion matrices that show box sorting results for each model, as seen in Figure 14. Model 1 in Figure 14(a) shows corresponds to two tree networks and model 2 in Figure 14(b) to the ouroboro type, the classes correspond to correct labeling of each type of box A=1, B=2 and C=3. Table 3 illustrates statistical results of confusion matrix metrics of each model used, it is evident that model 1 (tree) has a higher precision, however, both models present a high precision that exceeds 90%. There is a significant difference in the classification and localization time of each model, for model 1 it is 6.8565 seconds, while model 2 (ouroboro) is 4.884 seconds, almost two seconds more.



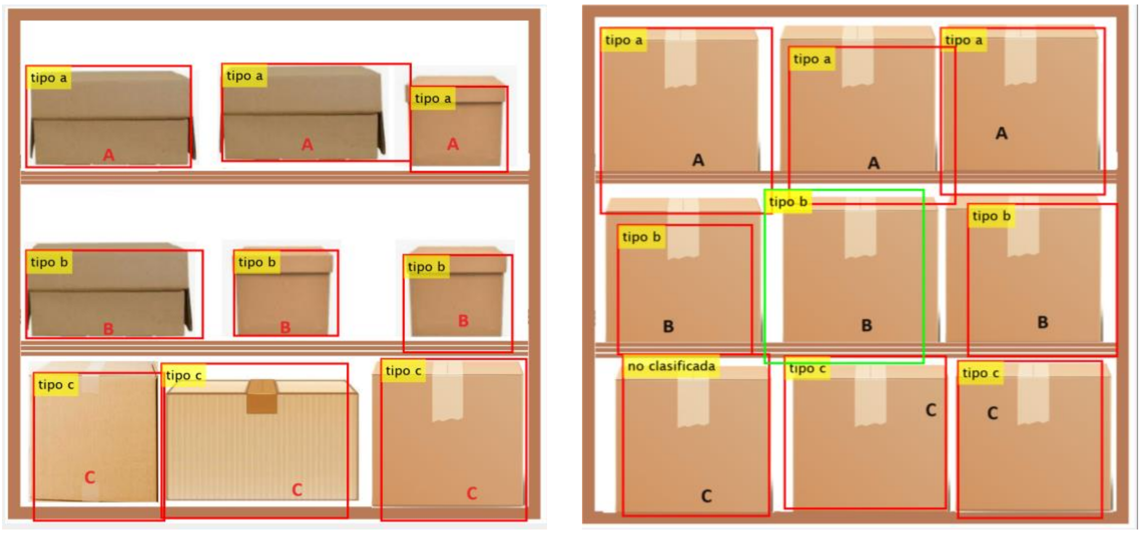
(a) (b)

Figure 11. Classification error: (a) misclassification and (b) network detections error



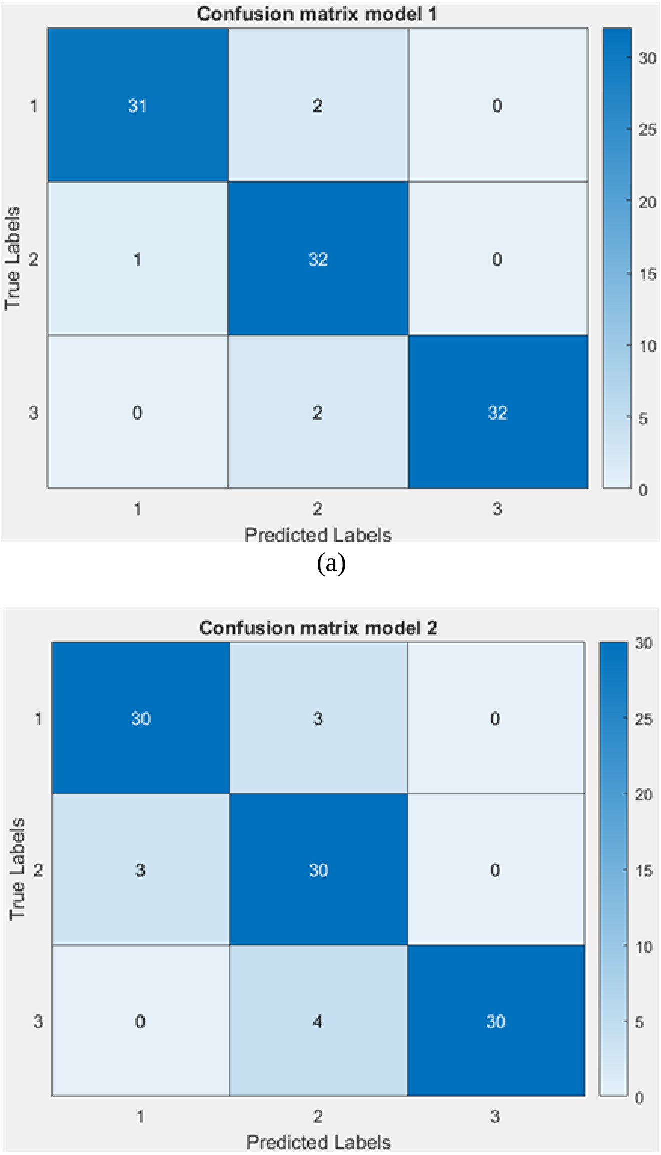
(a) (b)

Figure 12. Detection error due to size (a) 224×224 pixels and (b) 220×220 pixels



(a) (b)

Figure 13. Tree structure detection (a) correct classification and (b) classification without detection



(a)

(b)

Figure 14. Confusion matrices (a) tree model and (b) ouroboro model

Table 3. Confusion matrix metrics

Parameter	Tree model (1)	Ouroboro model (2)
Precision	0.96875	0.90909
Recall	0.93939	0.90909
F1-score	0.95385	0.90909

In general, main false positives that occur are due to changes in resolution of letters used as labels to discriminate the type of box. The reduction in resolution, as mentioned, generates a loss of characteristics that, in this case, causes letters to be confused mainly with B, as indicated by the confusion matrices. This case is carried out in rectangular boxes that present a greater loss of characteristics when they are resized to the size of network input. As future research, the construction of algorithms is proposed that allow not only the identification of boxes and their types, but also achieve the interpretation of the typical images used as guidance on the handling that should be given to the boxes for their manipulation and subsequently the interpretation of more detailed labels located on products. Progress could also be made towards expanding the scope by considering the handling of boxes and products by robotic agents for the preparation of orders.

4. CONCLUSION

It is concluded that the two solutions designed are fully functional and allow the objective of locating and identifying products stored in labeled boxes on shelves to be met with an accuracy greater than 90% and with processing times of less than 7 seconds, in both cases, which for an application in a real environment is a manageable response time. Despite existing differences in training times, for practical purposes these do not turn out to be relevant, as are precision and agility in response time as criteria for selecting the most convenient model in light of end user's objectives, which does allow us to conclude which model will be most convenient in application environment. It is evident that, although computational cost is higher when using two tree network architectures, by doubling use of resources, precision achieved is greater, due to specialization of the pattern learning of each network. Where it can be concluded that temporal accumulation can favor use of the ouroboro architecture, since for each execution two seconds advantage it presents over the tree model must be added.

ACKNOWLEDGEMENTS

Product derived from the research project titled "Mejoramiento de procesos de control de inventario y alistamiento de pedidos en el marco de la industria 4.0" INV-ING-3970 financed by the vice-rector for research of the Universidad Militar Nueva Granada, year 2024.

REFERENCES

[1] F. E. Habib, H. Bnouachir, M. Chergui, and A. Ammoumou, "Industry 4.0 concepts and implementation challenges: literature review," *2022 9th International Conference on Wireless Networks and Mobile Communications, WINCOM 2022*, 2022, doi: 10.1109/WINCOM55661.2022.9966456.

[2] X. Xin, S. L. Keoh, M. Sevegnani, M. Saerbeck, and T. P. Khoo, "Adaptive model verification for modularized industry 4.0 applications," *IEEE Access*, vol. 10, pp. 125353–125364, 2022, doi: 10.1109/ACCESS.2022.3225399.

[3] L. L. Yadla, R. A. Mudragada, H. Poka, and T. Vignesh, "Smart manufacturing in industries using internet of things," *2023 International Conference on Computer Communication and Informatics, ICCCI 2023*, 2023, doi: 10.1109/ICCCI56745.2023.10128206.

[4] G. Wang, D. Li, and H. Song, "A formal analytical framework for iot-based plug-and play manufacturing system considering product life-cycle design cost," *IEEE Transactions on Industrial Informatics*, vol. 19, no. 2, pp. 1647–1654, 2023, doi: 10.1109/TII.2022.3192681.

[5] B. Wu, "Motion control algorithm for automatic welding of complex intersecting line joints based on deep learning," *2023 International Conference on Mechatronics, IoT and Industrial Informatics, ICMIII 2023*, pp. 352–356, 2023, doi: 10.1109/ICMIII58949.2023.00073.

[6] S. Vaddadi, V. Srinivas, N. A. Reddy, H. Girish, D. Rajkiran, and A. Devipriya, "Factory inventory automation using industry 4.0 technologies," *2022 IEEE IAS Global Conference on Emerging Technologies, GlobConET 2022*, pp. 734–738, 2022, doi: 10.1109/GlobConET53749.2022.9872416.

[7] S. Dong, P. Wang, and K. Abbas, "A survey on deep learning and its applications," *Computer Science Review*, vol. 40, 2021, doi: 10.1016/j.cosrev.2021.100379.

[8] S. Albawi, T. A. Mohammed, and S. Al-Zawi, "Understanding of a convolutional neural network," *Proceedings of 2017 International Conference on Engineering and Technology, ICET 2017*, vol. 2018, pp. 1–6, 2017, doi: 10.1109/ICEngTechnol.2017.8308186.

[9] D. Erhan, C. Szegedy, A. Toshev, and D. Anguelov, "Scalable object detection using deep neural networks," *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 2155–2162, 2014, doi: 10.1109/CVPR.2014.276.

[10] H. T. Do and V. C. Pham, "Deep learning based goods management in supermarkets," *Journal of Advances in Information Technology*, vol. 12, no. 2, pp. 164–168, 2021, doi: 10.12720/jait.12.2.164-168.

[11] A. Börold, M. Teucke, A. Rust, and M. Freitag, "Deep learning-based object recognition for counting car components to support handling and packing processes in automotive supply chains," *IFAC-PapersOnLine*, vol. 53, pp. 10645–10650, 2020, doi: 10.1016/j.ifacol.2020.12.2828.

[12] N. D. Nguyen, T. Do, T. D. Ngo, and D. D. Le, "An evaluation of deep learning methods for small object detection," *Journal of Electrical and Computer Engineering*, vol. 2020, 2020, doi: 10.1155/2020/3189691.

[13] W. Ouyang *et al.*, "DeepID-Net: object detection with deformable part based convolutional neural networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 7, pp. 1320–1334, 2017, doi: 10.1109/TPAMI.2016.2587642.

[14] R. Duan, H. Deng, M. Tian, Y. Deng, and J. Lin, "SODA: A large-scale open site object detection dataset for deep learning in construction," *Automation in Construction*, vol. 142, 2022, doi: 10.1016/j.autcon.2022.104499.

[15] A. Sharma, T. Mishra, J. Kukade, A. Golwalkar, and H. Tomar, "Object detection using TensorFlow," *ICT Analysis and Applications*, pp. 343–352, 2023, doi: 10.1007/978-981-99-6568-7_31.

[16] F. Pilati *et al.*, "Operator 5.0: enhancing the physical resilience of workers in assembly lines," *2023 IEEE International Workshop on Metrology for Industry 4.0 and IoT, MetroInd4.0 and IoT 2023*, pp. 177–182, 2023, doi: 10.1109/MetroInd4.0IoT57462.2023.10180145.

[17] Y. W. Chan, T. H. Huang, Y. T. Tsan, W. C. Chan, C. H. Chang, and Y. Te Tsai, "The risk classification of ergonomic musculoskeletal disorders in work-related repetitive manual handling operations with deep learning approaches," *2020 International Conference on Pervasive Artificial Intelligence, ICPAI 2020*, pp. 268–271, 2020, doi: 10.1109/ICPAI51961.2020.00057.

[18] K. He, X. Zhang, S. Ren, and J. Sun, "Identity mappings in deep residual networks," *Computer Vision – ECCV 2016 (ECCV 2016)*, pp. 630–645, 2016, doi: 10.1007/978-3-319-46493-0_38.

[19] F. Ramzan *et al.*, "A deep learning approach for automated diagnosis and multi-class classification of alzheimer's disease stages using

resting-state fMRI and residual neural networks," *Journal of Medical Systems*, vol. 44, no. 2, 2020, doi: 10.1007/s10916-019-1475-2.

[20] T. N. V. S. Praveen, D. Sivathmika, G. Jahnavi, and J. Bolledu, "An in-depth exploration of ResNet-50 for complex emotion recognition to unraveling emotional states," *2023 International Conference on Advancement in Computation and Computer Technologies, InCACCT 2023*, pp. 322–326, 2023, doi: 10.1109/InCACCT57535.2023.10141774.

[21] D. R. Soumya, D. L. K. Reddy, A. Nagar, and A. K. Rajpoot, "Enhancing brain tumor diagnosis: utilizing ResNet-101 on MRI images for detection," *ViTECoN 2023 - 2nd IEEE International Conference on Vision Towards Emerging Trends in Communication and Networking Technologies, Proceedings*, 2023, doi: 10.1109/ViTECoN58111.2023.10157378.

[22] G. Kaur, N. Sharma, R. Chauhan, S. Kukreti, and R. Gupta, "Eye disease classification using ResNet-18 deep learning architecture," *2023 2nd International Conference on Futuristic Technologies, INCOFT 2023*, 2023, doi: 10.1109/INCOFT60753.2023.10425690.

[23] A. Chaudhuri, "Hierarchical modified fast R-CNN for object detection," *Informatica*, vol. 45, no. 7, pp. 67–81, 2021, doi: 10.31449/inf.v45i7.3732.

[24] S. Dahiya, T. Gulati, and D. Gupta, "Performance analysis of deep learning architectures for plant leaves disease detection," *Measurement: Sensors*, vol. 24, 2022, doi: 10.1016/j.measen.2022.100581.

[25] A. Almalaq and G. Edwards, "A review of deep learning methods applied on load forecasting," *16th IEEE International Conference on Machine Learning and Applications, ICMLA 2017*, vol. 2017, pp. 511–516, 2017, doi: 10.1109/ICMLA.2017.0-110.

BIOGRAPHIES OF AUTHORS

Anny Astrid Espitia Cubillos performed her undergraduate studies in Industrial Engineering in the Universidad Militar Nueva Granada in 2002 and M.Sc. in Industrial Engineering from the Universidad de Los Andes in 2006. She is an Associate Professor on Industrial Engineering Program at Universidad Militar Nueva Granada, Bogotá, Colombia. He can be contacted at email: anny.espitia@unimilitar.edu.co.

Robinson Jiménez-Moreno is an Electronic Engineer graduated from Universidad Distrital Francisco José de Caldas in 2002. He received a M.Sc. in Engineering from Universidad Nacional de Colombia in 2012 and Ph.D. in Engineering at Universidad Distrital Francisco José de Caldas in 2018. His current working as assistant professor of Universidad Militar Nueva Granada and research focuses on the use of convolutional neural networks for object recognition and image processing for robotic applications such as human-machine interaction. He can be contacted at email: robinson.jimenez@unimilitar.edu.co.

Esperanza Rodríguez Carmona was born in Pereira (Colombia). She performed her undergraduate studies in Mechanical Engineering (1997) from the Universidad Tecnológica de Pereira (UTP) and Master's degree in university teaching from the Universidad de La Salle, Bogotá, Colombia. She is an Associate Professor on Industrial Engineering program at Universidad Militar Nueva Granada, Bogotá, Colombia. He can be contacted at email: esperanza.rodriguez@unimilitar.edu.co.