WWW.IJRPR.COM

International Journal of Research Publication and Reviews

Journal Homepage: www.ijrpr.com ISSN 2582-7421

Python-Based Color and Shape Sorting in Coppelia Sim: An OpenCV and Random Forest Case Study

Mălina-Alexandra Bolovan*, Ștefan-Alexandru Lupșa, Cosmin Alin Baltag

National University of Science and Technology POLITEHNICA Bucharest, Bucharest 060042, Romania
e-mail address: malina.bolovan@stud.fiir.upb.ro

DOI : https://doi.org/10.55248/gengpi.07.1026.35014

A B S T R A C T

This paper presents a Python-based robotic sorting application developed in CoppeliaSim. The system includes two UR5 robotic manipulators, conveyor belts, and computer vision for identifying objects based on color and shape. Two classification methods were compared: conventional OpenCV image processing and Random Forest machine learning. The evaluation used a dataset of 170 images containing four object categories, with 34 images selected for testing. Random Forest achieved 100% classification accuracy, while OpenCV achieved 67.65%. Processing performance was also examined using two recorded simulation sessions, each containing 20 objects. OpenCV showed a lower average inference latency of 1.63 ms, compared with 63.66 ms for Random Forest. The results show that Random Forest provided better classification accuracy on the selected images, while OpenCV required less processing time. The study offers a simple comparison of two classification methods for simulated robotic sorting.

Keywords: CoppeliaSim; Python; Robotic Sorting; Computer Vision; OpenCV; Random Forest.

1. Introduction

Industrial automation increasingly integrates robotic manipulators, computer vision, and intelligent software to support flexible manufacturing operations. Vision-based robotic systems can recognize objects and coordinate manipulation tasks using information extracted from digital images [1]. Robotic sorting, object identification, and machine learning (ML) may all function together in one cohesive framework that is based on Python [2]. Automated systems may be better adjusted to various object properties with the use of sophisticated visual processing [3].

Automated sorting applications rely on accurate item categorization. Without human assistance, computer vision systems can detect important visual qualities including form, color, and spatial location. Applications for vision-based safety monitoring have shown how virtual robotic workstations may be interpreted by image processing [4]. Additional opportunities for automated categorization and decision-making are presented by ML approaches [5]. Offline AI applications have also used image-based recognition, proving that it is possible to combine visual processing with local software execution [6]. Nevertheless, processing time and recognition performance must be considered when choosing a classification approach [7].

The use of robotic simulation platforms allows for the development and evaluation of automated operations in a controlled environment prior to their actual deployment. Integrating external algorithms with simulated industrial manipulators has been shown via Python-based planning apps [8]. A number of recent research have investigated the use of artificial intelligence approaches for robotic manipulation and object categorization [9]. The representation of robotic workspaces influences how movements and operational constraints can be handled in simulation [10]. Graph-based planning studies have shown that different geometric representations can lead to different computational and planning characteristics [11]. Such considerations are relevant when robotic manipulators must operate alongside conveyors and repeatedly transfer objects between predefined locations.

The performance of automated sorting systems also depends on the coordination of perception, movement, and control. Geometric continuity has been investigated as an important characteristic of industrial robot trajectories [12]. Vision-based robotic grasping research has demonstrated how visual information can support manipulation decisions [13]. Optimization methods have also been applied to articulated robot trajectories to improve movement efficiency [14]. The prediction of human mobility in shared workplaces has also been investigated in research on robot navigation [15]. Coordinated software and robotic processes are more important in general, as these examples show.

Integrating several computing components into a single operating system is another essential aspect. One further use of vision-based teleoperation is the integration of robotic arm control with human hand tracking [16]. Efficient software-based engineering systems and artificial intelligence research has also highlighted the need of data- and application-specific evaluation of computational approaches [17]. In order to detect production flaws in photos, similar industrial inspection apps have employed YOLO-based techniques [18]. Changes like this show how important visual data is becoming for automated engineering choices.

Even though robotic automation has come a long way, complicated deep learning frameworks aren't always necessary for basic categorization issues. Research on ML in software engineering has also shown the importance of model selection and input features for different applications [19]. Previous comparative work on mathematical prediction methods has similarly highlighted that computational approaches should be selected according to the characteristics of the available data and the intended task [20]. For conveyor-based robotic sorting, a direct comparison between conventional image processing and a ML classifier can therefore provide useful evidence regarding classification performance and computational requirements.

This paper presents a Python-based robotic sorting system developed in CoppeliaSim using two UR5 robotic manipulators, conveyor belts, image processing, and automated object handling. Two approaches are evaluated: conventional OpenCV image processing and Random Forest classification. The comparison considers object recognition accuracy, processing latency, and recorded sorting-session duration. The objective is to identify the advantages and limitations of both approaches within the same simulated robotic architecture, providing a practical evaluation of lightweight visual classification methods for automated sorting applications.

2. Materials and Methods

2.1 Robotic Sorting System Architecture

Software applications, computational methodologies, and interactive control mechanisms are all part of what makes up current automated system development. Both the structuring of software development processes and flexible software applications for learning have been the subject of prior study [21, 22]. When it comes to industrial automation, where many software modules need to share data while the system is running, same architectural ideas might be used.

Python and the ZeroMQ Remote API were used for remote control of the robotic sorting system that was designed in this project and was implemented in CoppeliaSim. There are two sorting stations, five conveyor belts, a vision sensor, proximity sensors, and two UR5 robotic manipulators in the virtual manufacturing line. Color and form are used to identify objects before they are moved to their respective sorting areas. Fig. 1 shows the general layout of the robotic sorting system that was used in the simulation. It includes the conveyor system, the robotic manipulators, and the stations that handled the objects.

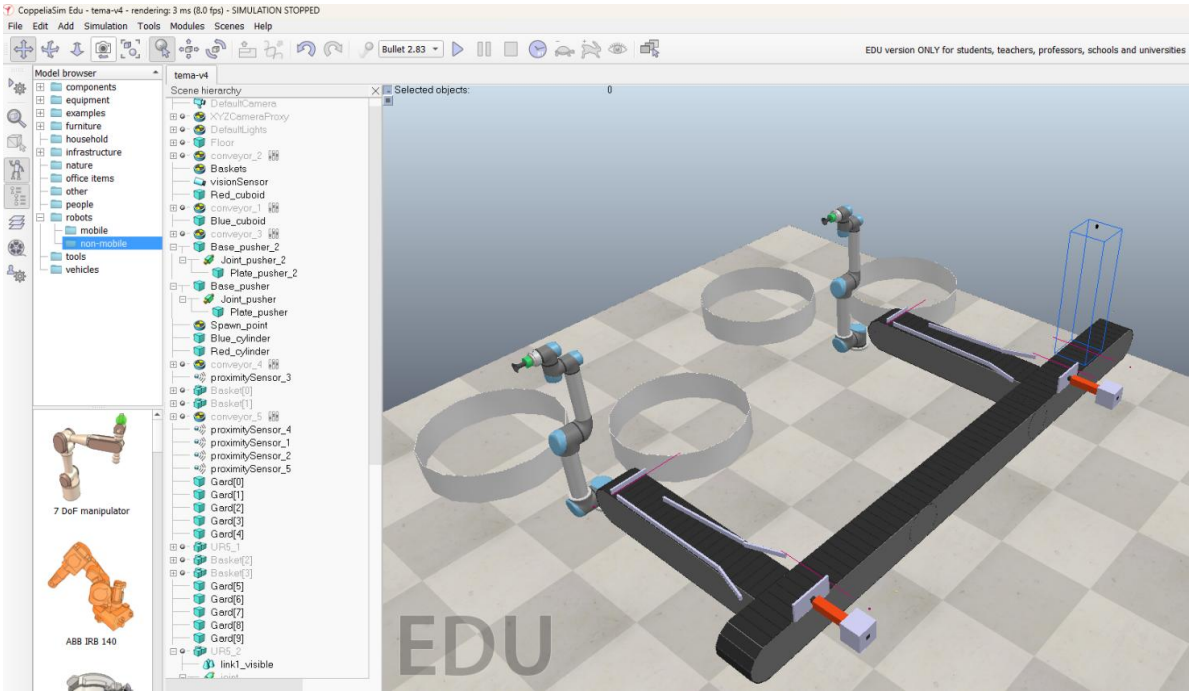


Fig. 1. Overall architecture of the automated robotic sorting system developed in CoppeliaSim.

The software architecture consists of independent modules for object generation, visual processing, classification, conveyor control, robot manipulation, and performance monitoring. The integration of several processing components has also been investigated in multimodal intelligent interfaces [23,24], although the present application uses camera-based classification rather than human physiological inputs. A PyQt6 graphical interface allows the operator to control the simulation and select the classification approach.

Efficient execution also depends on coordinating software components and maintaining communication between processing tasks. Previous research has explored multithreaded software applications [25], while other studies have examined tools for monitoring complex software systems [26]. In the proposed system, the graphical interface and simulation control are separated to maintain responsiveness during robotic execution.

The development of intelligent engineering applications can also be viewed within the broader context of Industry 5.0 and digital transformation [33]. Research has examined how organizations prepare for new management approaches during technological changes [34]. At the application level, sensors and control software have been integrated into home automation systems [35] and embedded prosthetic devices [36]. ML has also been applied to resource allocation and decision-making in engineering projects [37]. Although these studies address different problems, they show how software, data processing, and control functions can be combined in practical applications. The present study applies a similar general approach to a simulated robotic sorting system.

2.2 Object Classification Methods

Two classification approaches were implemented: conventional OpenCV image processing and Random Forest ML:

- The conventional method converts the acquired images into HSV color space and applies predefined thresholds to distinguish red and blue objects. Shape identification is based on contour analysis using the extent parameter, defined as the ratio between contour area and bounding rectangle area. A threshold of 0.88 is used to distinguish cubes from cylinders.
- The Random Forest method uses four numerical characteristics extracted from the images: mean hue, mean saturation, circularity, and extent. These characteristics are provided to a classifier consisting of 100 decision trees.

Using ML in various decision-support and prediction settings has shown how crucial it is to choose the right input variables and assessment methods [27,28]. Rather than robotic classification, these research focus on educational prediction, yet they nevertheless demonstrate supervised learning in action using structured datasets.

Data properties and analytical goals can have a role in the choice of computational techniques. Prediction methods have been investigated in supervised learning applications [29], while ML has also been used in software project management [30]. In the present study, this principle motivates the comparison between a deterministic image-processing method and a supervised classification model, without assuming that findings from other prediction domains directly transfer to robotic sorting.

Data processing and classification techniques have also been investigated in other intelligent applications, including brain-to-text prediction systems [38]. There have been developments in software-based decision-support systems for evaluating employee performance using structured indicators [40] and the use of supervised learning to bolster academic human resources prediction [39]. All of these uses case show how computational techniques may be used to recorded data in order to make predictions, classifications, or judgments, even when the input data and goals are different.

2.3 Dataset and Experimental Configuration

The image dataset contains 170 samples distributed into four categories: blue cubes (40), red cubes (35), blue cylinders (51), and red cylinders (44). For the classification evaluation, the 170 images were ordered alphabetically by their file paths before being divided into 80% training data and 20% testing data, using a fixed random state of 42. Both classification methods were evaluated using the same 34 test images.

Both methods were integrated into the same virtual sorting architecture. The experimental configuration includes object generation, image acquisition, classification, conveyor transportation, robotic manipulation, and performance recording.

Monitoring and visualizing operational information are important aspects of software-assisted evaluation. Previous research has presented visualization methods for identifying variations in recorded digital activity [31,32]. Although developed for different applications, these studies illustrate the general usefulness of organizing recorded information into interpretable indicators.

The experimental evaluation uses two recorded sorting sessions containing 20 object-processing records for each classification method. The collected information includes classification latency and session duration. In addition, the archived image dataset allows classification accuracy to be evaluated separately. These measurements provide the basis for comparing conventional image processing and Random Forest classification within the proposed robotic sorting environment.

Engineering studies use different methods to evaluate system performance. Numerical analysis has been used to investigate the dynamic behavior of bionic mechanical systems [41], while other research has compared numerical and experimental results for structural components [42]. Multicriteria methods have also been applied to the evaluation of material recovery alternatives [43]. In applications involving personal information, data use and governance have been examined as additional aspects of system evaluation [44]. These studies address different evaluation needs. In the present work, the analysis is limited to image classification accuracy and recorded processing times in a simulated robotic sorting system.

3. Results

The two classification methods were evaluated using the available image dataset and the recorded CoppeliaSim sessions. The evaluation focused on classification accuracy and processing time. Table 1 presents the classification results obtained using 34 test images from the dataset of 170 images.

Table 1. Classification results for OpenCV and Random Forest.

Indicator	OpenCV	Random Forest
Test images	34	34
Correct classifications	23	34
Accuracy (%)	67.65	100.00

Random Forest correctly classified all 34 test images, while OpenCV correctly classified 23 images. The OpenCV method correctly identified all 18 cylinders, but only 5 of the 16 cubes. These results show that Random Forest performed better on the selected dataset. However, the number of test images was relatively small.

The processing performance was also examined using the recorded CoppeliaSim sessions. Each session contained 20 object-processing records, as presented in Table 2.

Table 2. Processing performance during the recorded sorting sessions.

Indicator	OpenCV	Random Forest
Recorded objects	20	20
Mean inference latency (ms)	1.63	63.66
Minimum latency (ms)	1.29	59.40
Maximum latency (ms)	3.41	87.86
Session duration (s)	222.40	238.66

OpenCV was considerably faster than Random Forest. Its average inference latency was approximately 39 times lower. The recorded session was also 16.26 seconds shorter with OpenCV. However, the total session duration included conveyor movement and robotic operations, not only image classification.

Overall, the results show a difference between classification accuracy and processing speed. Random Forest achieved higher accuracy on the test images, while OpenCV required less processing time. The evaluation was limited to four object categories in a simulated environment. Some test images had the same extracted features as training images, which may have influenced the Random Forest results. In addition, the recorded sessions did not provide independent confirmation of correct object placement. Therefore, these findings represent a preliminary comparison of two simple classification methods for robotic sorting.

4. Discussion and Conclusions

The results show that both classification methods can be used in a simulated robotic sorting system, but they offer different advantages. Random Forest achieved better accuracy on the selected images, while OpenCV required less processing time. This suggests that the choice of classification method depends on the needs of the application.

The study was limited to four object categories and a relatively small image dataset. The system was tested in CoppeliaSim, without validation on physical robots. Future work may include more object types, different lighting conditions, and tests using real robotic equipment.

In conclusion, the developed Python application allowed the comparison of two classification approaches within the same simulated sorting system. The results provide a useful starting point for further work on simple vision-based robotic applications.

References

1. Hong, F., Tay, D. W. L., & Ang, A. (2023). Intelligent pick-and-place system using MobileNet. Electronics, 12(3), 621. https://doi.org/10.3390/electronics12030621
2. Drăgoi, M.-V., Cristoiu, C. A., Nechita, R.-M., Navligu, B.-C., & Verdete, B.-M. (2026). A Python-based framework for learning-from-demonstration in robotic object sorting: Comparative evaluation of lightweight classifiers. Applied Sciences, 16(10), 5107. https://doi.org/10.3390/app16105107
3. Kee, E., Chong, J. J., Choong, Z. J., & Lau, M. (2023). Development of smart and lean pick-and-place system using EfficientDet-Lite for custom dataset. Applied Sciences, 13(20), 11131. https://doi.org/10.3390/app132011131

4. Cristoiu, C. A., Drăgoi, M.-V., Cojocaru, A., & Spânu, P. (2026). A vision-based software safety monitoring tool for operators in RoboDK robotic cells: A simulation-based proof-of-concept study using workspace masks and image processing. Technologies, 14(6), 373. https://doi.org/10.3390/technologies14060373

5. Glučina, M., Anđelić, N., Lorencin, I., & Car, Z. (2023). Detection and classification of printed circuit boards using YOLO algorithm. Electronics, 12(3), 667. https://doi.org/10.3390/electronics12030667

6. Drăgoi, M.-V., Nisipeanu, I., Puiu, R.-A., Tache, F.-G., Spiridon-Mocioacă, T.-M., Hank, A., & Cristoiu, C. (2025). An inclusive offline learning platform integrating gesture recognition and local AI models. Biomimetics, 10(10), 693. https://doi.org/10.3390/biomimetics10100693

7. Da Rold, A., Furiato, M., Zaki, A. M. A., Carnevale, M., & Giberti, H. (2023). Deep learning-based robotic sorter for flexible production. Procedia Computer Science, 217, 1579–1588. https://doi.org/10.1016/j.procs.2022.12.358

8. Cristoiu, C. A., Drăgoi, M.-V., Nechita, R.-M., Verdete, B.-M., Dudici, C. L., & Cusma, C. N. (2026). Python software application for obstacle-avoiding path planning in RoboDK using free space graph and robot level validation. Applied Sciences, 16(10), 4786. https://doi.org/10.3390/app16104786

9. Kluziak, S., & Kohut, P. (2025). Development of a UR5 cobot vision system with MLP neural network for object classification and sorting. Information, 16(7), 550. https://doi.org/10.3390/info16070550

10. Cristoiu, C.-A., Drăgoi, M.-V., & Georgescu, V.-C. (2026). Comparative analysis of Cartesian, cylindrical and spherical grids in a graph-based obstacle-avoidance planner for industrial robots. Applied Sciences, 16(12), 6189. https://doi.org/10.3390/app16126189

11. Ji, Y., Zhou, C., Lin, R., & Liu, Q. (2025). Path planning strategy for a space relocatable robotic manipulator based on improved GBNN algorithm. Robotics and Autonomous Systems, 187, 104939. https://doi.org/10.1016/j.robot.2025.104939

12. Cristoiu, C. A., Drăgoi, M.-V., Nechita, R.-M., Stoian, B. N., Dudici, C. L., Puiu, R.-A., & Petrea, G. (2026). The impact of geometric continuities (C1, C2, and C3) on the trajectories of industrial robots. Technologies, 14(3), 176. https://doi.org/10.3390/technologies14030176

13. Chen, Y.-L., Cai, Y.-R., & Cheng, M.-Y. (2023). Vision-based robotic object grasping—A deep reinforcement learning approach. Machines, 11(2), 275. https://doi.org/10.3390/machines11020275

14. Cristoiu, C. A., Drăgoi, M.-V., Ivan, A. M., Nechita, R.-M., Grecu, I., Puiu, R.-A., Petrea, G., & Popescu, E. (2025). Optimization method based on the minimum action principle for trajectory length of articulated manipulators. Technologies, 13(11), 490. https://doi.org/10.3390/technologies13110490

15. Dik, C., Emmanouilidis, C., & Duqueroie, B. (2024). Graph network-based human movement prediction for socially-aware robot navigation in shared workspaces. Neural Computing and Applications, 36, 21743–21759. https://doi.org/10.1007/s00521-024-10369-x

16. Drăgoi, M.-V., Frimu, A.-V., Postelnicu, A., Puiu, R.-A., Petrea, G., & Hank, A. (2026). Interactive teleoperation of an articulated robotic arm using vision-based human hand tracking. Biomimetics, 11(2), 151. https://doi.org/10.3390/biomimetics11020151

17. Abbas, T., Rathore, S. A., Turki, A., Khan, S., Alghushairy, O., & Daud, A. (2025). Enhancing software engineering with AI: Innovations, challenges, and future directions. IET Software, 2025(1), 5691460. https://doi.org/10.1049/sfw2/5691460

18. Tălîngă, A.-M., Hadăr, A., Drăgoi, M.-V., Nisipeanu, I., Ali, H. A., & Suciu, C. P. (2024). YOLO-v8 in capturing imperfections generated by changing 3D printer parameters. U.P.B. Scientific Bulletin, Series C, 86(4), 253–266. https://www.scientificbulletin.upb.ro/rev_docs_arhiva/full72b_799902.pdf

19. Terlapu, P. V., Raju, K. K., Kumar, G. K., Rao, G. J., Kavitha, K., & Samreen, S. (2024). Improved software effort estimation through machine learning: Challenges, applications, and feature importance analysis. IEEE Access, 12, 138663–138701. https://doi.org/10.1109/ACCESS.2024.3457771

20. Drăgoi, M.-V., Afram, R. M., Alzubaidi, A. A. A. M. H., Jiga, G. G., Ali, H. A., Puiu, R.-A., Petrea, G., & Hank, A. (2026). Comparative analysis of interpolation and extrapolation methods in regression models. U.P.B. Scientific Bulletin, Series C, 88(1), 145–160. https://www.scientificbulletin.upb.ro/seria/c/arhiva?year=2026

21. Drăgoi, M.-V., Puiu, R.-A., Petrea, G., Burtea, C., Spiridon-Mocioacă, T.-M., Puiu, M., & Suciu, C. P. (2025). Improving the learning process for students. Personalizing learning through software applications. U.P.B. Scientific Bulletin, Series C, 87(1), 157–166. https://www.scientificbulletin.upb.ro/rev_docs_arhiva/full465_531870.pdf

22. Amaro, R., Pereira, R., & da Silva, M. M. (2025). Mapping DevOps capabilities to the software life cycle: A systematic literature review. Information and Software Technology, 177, 107583. https://doi.org/10.1016/j.infsof.2024.107583

23. Drăgoi, M.-V., Nisipeanu, I., Marin, I., Cristoiu, C. A., & Alexe, C.-G. (2026). Bio-inspired gaze and neural command fusion for assistive smartphone interaction. Biomimetics, 11(7), 514. https://doi.org/10.3390/biomimetics11070514

24. Dritsas, E., Trigka, M., Troussas, C., & Mylonas, P. (2025). Multimodal interaction, interfaces, and communication: A survey. Multimodal Technologies and Interaction, 9(1), 6. https://doi.org/10.3390/mti9010006

25. Drăgoi, M.-V., Maier, A. N., Alzubaidi, A. A. A. M. H., Suciu, C. P., Ali, H. A., Dobrescu, T. G., Hadăr, A., Puiu, R.-A., Petrea, G., & Avăsiloaie, A. P. (2025). Multithreading VPN application to monitor web traffic and unblock geoblocking websites. U.P.B. Scientific Bulletin, Series C, 87(4). https://www.scientificbulletin.upb.ro/static/pdfs/rez17b_486740.pdf

26. Giamattei, L., Guerriero, A., Pietrantuono, R., Russo, S., Malavolta, I., et al. (2024). Monitoring tools for DevOps and microservices: A systematic grey literature review. Journal of Systems and Software, 208, 111906. https://doi.org/10.1016/j.jss.2023.111906

27. Buzducea, C.-A., Drăgoi, M.-V., Cristoiu, C., Puiu, R.-A., Puiu, M., Petrea, G., & Navligu, B.-C. (2026). Machine learning in education: Predicting student performance and guiding institutional decisions. Education Sciences, 16(1), 76. https://doi.org/10.3390/educsci16010076

28. Oppong, S. O. (2023). Predicting students' performance using machine learning algorithms: A review. Asian Journal of Research in Computer Science, 16(3), 128–148. https://doi.org/10.9734/ajrcos/2023/v16i3351

29. Jiang, B., Zhu, X., Tian, X., Yi, W., & Wang, S. (2024). Integrating interpolation and extrapolation: A hybrid predictive framework for supervised learning. Applied Sciences, 14(15), 6414. https://doi.org/10.3390/app14156414

30. Mahdi, M. N., Mohamed Zabil, M. H., Ahmad, A. R., Ismail, R., Yusoff, Y., Cheng, L. K., Mohd Azmi, M. S. B., Natiq, H., & Happala Naidu, H. (2021). Software project management using machine learning technique—A review. Applied Sciences, 11(11), 5183. https://doi.org/10.3390/app11115183

31. Drăgoi, M., Buzducea, C., Marin, A., & Boanță, L. (2026). Assessment pressure mapper for visualizing peaks in digital evaluation load. FAIMA Business & Management Journal, 14(1), 34–42. https://faimajournal.upb.ro/archive/RevistaFAIMA_Martie2026.pdf

32. Reszka, Ł., Sosnowski, J., & Dobrzyński, B. (2023). Enhancing software project monitoring with multidimensional data repository mining. Electronics, 12(18), 3774. https://doi.org/10.3390/electronics12183774

33. Drăgoi, M.-V., Cristoiu, C., & Tamaș, E. (2026). Industry 5.0 and sustainable business models in digital transformation. FAIMA Business & Management Journal, 14(3), 57–67. https://faimajournal.upb.ro/archive/RevistaFaima_11sept2026_final.pdf

34. Drăgoi, M.-V., Șerban, D. O., Cristoiu, C. A., Boanță, L. F., & Marin, A. (2026). Hybrid project management readiness in Romanian construction companies: An exploratory survey from a small construction-company context. Buildings, 16(19), 3877. https://doi.org/10.3390/buildings16193877

35. Drăgoi, M.-V., Nisipeanu, I., Frimu, A., Tălîngă, A.-M., Hadăr, A., Dobrescu, T. G., Suciu, C. P., & Manea, A. R. (2024). Real-time home automation system using BCI technology. Biomimetics, 9(10), 594. https://doi.org/10.3390/biomimetics9100594

36. Drăgoi, M.-V., Hadăr, A., Goga, N., Grigore, L. Ș., Ștefan, A., & Ali, H. A. (2023). Design and implementation of an EEG-based BCI prosthetic lower limb using Raspberry Pi 4. U.P.B. Scientific Bulletin, Series C, 85(3), 353–366. https://www.scientificbulletin.upb.ro/static/pdfs/full704_649223.pdf

37. Tache, F.-G., Drăgoi, M.-V., Puiu, R.-A., Petrea, G., & Pascaru, R.-C. (2026). Application in research projects on the innovation found in the field of engineering of polymer and composites products. Macromolecular Symposia, 415(3), e70379. https://doi.org/10.1002/masy.70379

38. Ali, H. A., Goga, N., Vasilateanu, A., Muslim, A. M., Ali, K. A., & Drăgoi, M.-V. (2022). English and Romanian brain-to-text brain-computer interface word prediction system. International Journal of Advanced Computer Science and Applications, 13(10). https://doi.org/10.14569/IJACSA.2022.0131003

39. Puiu, R.-A., Drăgoi, M.-V., Tache, F.-G., Spiridon-Mocioacă, T.-M., Petrea, G., & Hank, A. (2026). Interactive attrition prediction for academic human resources management using machine learning. U.P.B. Scientific Bulletin, Series C, 88(2), 217–232. https://www.scientificbulletin.upb.ro/static/pdfs/a2026nr2c_16_16653_Roxana_Puiu_2_pag.pdf

40. Tache, F.-G., Drăgoi, M.-V., Puiu, R.-A., Petrea, G., & Pascaru, R.-C. (2025). A decision-support tool for evaluating employee contribution toward organizational excellence. In International Conference of Management and Industrial Engineering (Vol. 12, pp. 345-352). https://doi.org/10.56177/12icmie2025.109

41. Drăgoi, M.-V., Hadăr, A., Goga, N., Baciu, F., Ștefan, A., Grigore, L. Ș., Gorgoteanu, D., Molder, C., & Oncioiu, I. (2023). Contributions to the dynamic regime behavior of a bionic leg prosthesis. Biomimetics, 8(5), 414. https://doi.org/10.3390/biomimetics8050414

42. Tălîngă, A.-M., Baciu, F., Hadăr, A., Jiga, G., Drăgoi, M.-V., & Ancuța, A. (2025). Numerical and experimental study of a sandwich beam subjected to a three points bending test. U.P.B. Scientific Bulletin, Series D, 87(3), 227–240. https://www.scientificbulletin.upb.ro/static/pdfs/fulld2b_804128.pdf

43. Nechita, R.-M., Deselnicu, D.-C., Băjenaru, V.-D., Istrițeanu, S.-E., Cristoiu, C., & Drăgoi, M.-V. (2026). Optimizing material recovery from photovoltaic waste: A performance ranking using hybrid BWM-PROMETHEE II. Sustainability, 18(8), 3750. https://doi.org/10.3390/su18083750

44. Drăgoi, M.-V., Puiu, R.-A., Petrea, G., Cristoiu, C. A., & Dumitrescu, C.-I. (2026). Willingness to allow educational data use for learning analytics in higher education: Trust and governance predictors: An exploratory study. Education Sciences, 16(4), 637. https://doi.org/10.3390/educsci16040637