International Information and Engineering Technology Association
Advancing the World of Information and Engineering

Revue d'Intelligence Artificielle
Vol. 37, No. 6, December, 2023, pp. 1407-1414
Journal homepage: http://iieta.org/journals/ria

Integration of the Faster R-CNN Algorithm for Waste Detection in an Android Application

Imam Husni Al Amin*, Dwiki Vio Setyadarma, Setyawan Wibisono

Faculty of Information Technology and Industry, Universitas Stikubank, Semarang 50241, Indonesia

Corresponding Author Email: imam@edu.unisbank.ac.id

https://doi.org/10.18280/ria.370604

Received: 29 June 2023
Revised: 11 September 2023
Accepted: 8 October 2023
Available online: 27 December 2023

Keywords:
android, faster R-CNN, machine learning, waste detection

ABSTRACT

The surge in global population and economic activity has precipitated a significant escalation in waste generation, with projections indicating potential daily global waste levels of 11 million tons by the century's conclusion. This intensification presents a formidable challenge requiring innovative solutions, one of which is the utilization of machine learning algorithms for automated waste categorization. This study explores the integration of the Faster R-CNN algorithm within an Android application, aimed at streamlining waste management through the identification and categorization of recyclables. The proliferation of smartphone usage—specifically Android applications—provides an accessible platform for mass education on waste management, thereby contributing to potential waste reduction. This study deployed a visual testing approach to evaluate the application's performance across diverse waste categories, including cardboard, glass, plastic, and paper. During each testing session, images of each waste type were captured and the objects were methodically rotated by approximately 20 degrees, enhancing the robustness of the machine learning model. The implementation of the Faster R-CNN algorithm within an Android environment, as exemplified in this study, has demonstrated noteworthy potential to revolutionize waste management. An impressive accuracy rate of 98.106% was achieved in the detection and classification of various waste types. Such a technological innovation can augment the efficiency of waste categorization and recycling efforts. Thus, the study contributes to the development of a more sustainable and environmentally responsible waste management system.

1. INTRODUCTION

The domain of deep learning, which represents an intriguing and highly impactful facet of artificial intelligence, holds paramount significance in contemporary times. This is owing to the fact that, in recent years, deep learning has heralded a plethora of significant breakthroughs in numerous domains such as cyber security [1, 2], medical imaging [3, 4], computer vision [5], and an assortment of others, thereby solidifying its position as a highly influential and dynamic field of study. The field of deep learning has undergone significant progressions in recent years, fundamentally altering our comprehension and approach to resolving a diverse range of intricate issues. These advancements have paved the way for breakthroughs in the domains of facial recognition [6], object detection [7], mask detection [8], machine translation [9], speech recognition [10], COVID-19 detection [11], and numerous others.

The generation of waste materials, a byproduct of human activities, has risen to a level of concerningly high proportions due to the surge in population growth and economic expansion. The perpetuation of practices that generate waste has led to an unprecedented amount of waste being produced at a global scale, with an estimated 3.3 million tons generated each day [12]. This unprecedented level of waste production has presented numerous challenges for countries around the world in their management and mitigation of this issue [13].

Projections indicate that, in the year 2100, the daily rate of waste generated globally could potentially escalate to a staggering 11 million tons [14]. The sheer volume of waste unceremoniously disposed of in landfills poses a multitude of grave environmental problems, including but not limited to soil, water, and air contamination, as well as negative impacts on the health of the human populace. Alarming reports suggest that the waste produced by the planet is culpable for an alarming 5% of the cumulative carbon emissions across the world, and should waste incineration methods be employed, it is inevitable that this percentage will increase. Nevertheless, should we manage to implement superior waste management systems, we could potentially make a significant dent in global CO_2 emissions, with a potential reduction of up to a noteworthy 15% [15].

The importance of effective management of Municipal Solid Waste (MSW) cannot be overstated, as it serves as a major factor in curbing the looming environmental threat posed by the ever-increasing trend of waste generation. The facilitation of environmental protection, conservation of natural resources, and mitigation of public health risks that proper waste management offers cannot be overemphasized [16]. Therefore, for effective MSW management to be achieved, adequate recognition and comprehension of the nature of MSW are of utmost importance, particularly with regard to the quantity generated and collected for treatment (via recycling and energy recovery) and disposal [17].

In the field of computer vision, Deep Convolutional Neural Networks (CNNs) have achieved remarkable success in various tasks. Particularly, for object detection, region-based CNN detection techniques have become the dominant paradigm. This area has been expanding at an unprecedented pace, as evidenced by the emergence of three generations of region-based CNN detection models, namely R-CNN [18], Fast R-CNN [19], and the latest Faster R-CNN [20], each of which has exhibited progressively superior precision and faster processing speed over the recent years. The Faster R-CNN, which constitutes the most recent iteration of region-based generic object detection techniques, has been able to exhibit noteworthy outcomes across several object detection benchmarks. In addition, it has served as the fundamental framework for the triumphant submission in the COCO detection challenge of 2015.

In their previous work Liu et al. [21], they utilized the Faster R-CNN deep learning framework in conjunction with the simplified VGG16 extraction network to successfully achieve precise identification of rock types. An accuracy rate of over 96% was obtained for single-type rock image recognition, while more than 80% accuracy for multi-type rock hybrid images was achieved. Subsequently, in a follow-up study [22], it was determined that the proposed COVID Faster R-CNN framework might effectively aid in the detection of COVID-19 cases through the examination of Chest X-Ray images, with an impressive accuracy of 97.36%.

In the research conducted by Nguyen et al. [23], the Faster R-CNN framework was employed to detect lung nodules in CT scans. The system that was proposed in that study was able to attain a remarkably high sensitivity of 95.64% at a rate of 1.72 false positives per scan. Moreover, the Competition Performance Metric (CPM) score of the proposed system was 88.2%, which clearly indicates that it surpassed other state-of-the-art detection techniques. The false positive reduction network that was employed in this research also achieved impressive results, with a sensitivity of 93.8%, specificity of 97.6%, and accuracy of 95.7%. Furthermore, in a different study Zhang et al. [24], the faster R-CNN method was utilized to detect cells in fecal images. The proposed method was able to achieve an average mean precision of 84% and a detection time of 723 m/s per sample for a total of 40,560 fecal images. The accuracy of the method proposed in the aforementioned study was also considerable, with a success rate of 94% in counting cattle in pastures and 92% in feedlots [25].

From the aforementioned contextual background, it becomes evident that the confluence of deep learning, waste management, and Faster R-CNN presents a compelling solution to address the challenges posed by the escalating waste crisis. This article specifically investigates the pivotal role of Faster R-CNN in the context of waste detection. Our research primarily centers on the application of Faster R-CNN within an Android-based waste type detection system, with the aim of transforming waste sorting by type, thereby significantly mitigating its detrimental environmental impact. Furthermore, this study explores the integration of deep learning technology as an advanced, environmentally friendly solution for waste sorting, emphasizing its indispensable utility in effectively addressing this critical issue.

2. METHODOLOGY

The comprehensive and rigorous framework of the research approach for identifying various waste types in the Android platform utilizing the state-of-the-art deep learning framework, namely TensorFlow, with the advanced Faster R-CNN algorithm, encompasses a multitude of crucial stages that are of paramount importance to ensure the validity and reliability of the outcomes.

The initial stage of this method involves the collection of an extensive and diverse dataset about the waste types under consideration, followed by the meticulous and intricate process of dataset annotation, whereby each image is thoroughly analyzed and labeled by expert annotators to facilitate the development of a robust and efficient model. Subsequently, the data preparation stage involves the cleaning and transformation of the dataset such that it is suitable for the model training process. The image preprocessing phase is a crucial step that involves a series of intricate operations such as image resizing, normalization, and enhancement, which are aimed at optimizing the performance of the model. The next stage in this method is the model training phase, wherein the deep neural network is trained on the annotated and preprocessed dataset in an iterative manner until an optimal and accurate model is attained. The model evaluation stage encompasses an in-depth and meticulous analysis of the model's performance based on various metrics such as precision, recall, and F1-score, which are crucial indicators of the model's efficacy. The testing and validation phase involves the testing and validation of the model on independent and unseen datasets to assess the generalizability and robustness of the model. Finally, the results of the proposed method are presented and analyzed in detail to provide valuable insights and observations. For a more elaborate and detailed understanding of the various stages of the proposed method, please refer to Figure 1 below.

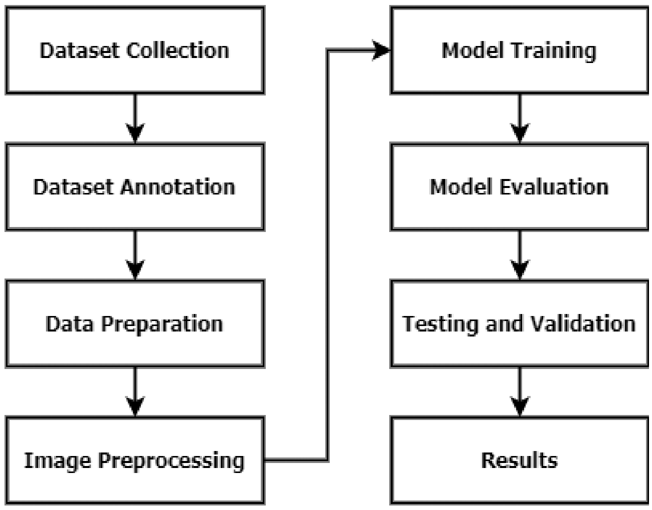

Figure 1. Stages of the research method

2.1 Dataset

The datasets, which were utilized in this particular study, encompass a grand total of 122 datasets that are inclusive of a diverse range of household waste, including but not limited to cardboard, glass bottles, glass, plastic, newspapers, old magazines, and even used plastic bottles. Once the datasets have been prepared, the subsequent step is to meticulously create a label for each individual dataset, one by one. Upon the completion of this meticulous labeling process, the datasets are then subjected to rigorous training via the algorithm from the

Faster R-CNN method. For a more comprehensive understanding of the intricacies of the aforementioned dataset, kindly refer to Figure 2.

Figure 2. Waste dataset

2.2 Faster R-CNN

The Faster Region-based Convolutional Neural Network (R-CNN) [26], a prominent two-stage architecture for object detection, has garnered considerable attention in the field of computer vision. This system, comprising a Regional Proposal Network (RPN) and a Fast Region by Feature Artificial Neural Network (Fast R-CNN), can be conceptualized as a complex entity that leverages the capabilities of both components to achieve superior performance. Specifically, the RPN is responsible for replacing the Selective Search algorithm [27] of Fast R-CNN, thereby enhancing the overall efficiency and accuracy of the system.

The Faster R-CNN model is comprised of a pair of modules, namely the Region Proposal Network (RPN) and the detector. It is pertinent to note that these networks share a similar convolutional backbone for feature extraction, which is generally pre-trained for classification and personalized for object detection. Moreover, the RPN and detector possess two heads each, one for predicting bounding boxes, which is referred to as the regression head, and the other for classification. It is imperative to highlight that the aforementioned components are critical in facilitating the efficiency of the Faster R-CNN model.

The Region Proposal Network (RPN) is a core component of Faster R-CNN, consisting of a classifier and a regressor. Anchors, predefined bounding boxes, are central to its operation, serving as reference points in a sliding window approach. For instance, the ZF Model uses 256-dimensional anchors, while VGG-16 employs 512-dimensional ones. The classifier determines objectness scores for these anchors, distinguishing objects from the background. The regressor fine-tunes anchor positions and sizes, improving object localization. Parameters like scale and aspect ratio are essential for adapting anchors to different image characteristics. Overall, RPN efficiently generates candidate object regions, enhancing object detection's speed and accuracy.

2.3 Faster R-CNN algorithm architecture

Faster R-CNN, an advanced and superior object detection system, has gained a prominent place in the field of deep learning owing to its remarkable capabilities and performance. One of the most significant benefits of this cutting-edge system is its impressive detection speed, which surpasses that of its forerunners, namely R-CNN and Fast R-CNN, as acknowledged by Ren et al. [26]. The superiority of Faster R-CNN can be attributed to its highly efficient and innovative architecture that has revolutionized the realm of object detection, enabling faster testing time per image.

The report in question posits that Faster R-CNN, a convolutional neural network model widely employed in the field of object detection, boasts a significantly expedited testing time per image in contrast to its predecessor R-CNN, with the former demonstrating an approximate acceleration of 250 times. This marked discrepancy can be attributed to the innovative incorporation of a Region Proposal Network (RPN) layer within the Faster R-CNN architecture, which functions to generate proposals of regions that possess the potential to harbor objects with admirable efficiency. The RPN layer facilitates swifter object detection by substantially diminishing the number of proposals that necessitate comprehensive evaluation.

Furthermore, it is worth noting that Faster R-CNN has been reported to possess an immensely expedited testing time per image, which is estimated to be a staggering 25 times greater than that of its predecessor, Fast R-CNN. This significant disparity can be attributed to the integration of two novel layers, namely the Region Proposal Network (RPN) layer and the object detection layer, into the Faster R-CNN framework. The inclusion of these layers has led to a much more streamlined and efficient detection process, which is now seamlessly integrated into the architecture of the Faster R-CNN model.

The augmented velocity of detection that is made possible by the Faster R-CNN algorithm is a significant aspect that bolsters its potential for utilization in real-world, time-sensitive applications or tasks that necessitate rapid responsiveness. Due to the significantly quicker test times enabled by the Faster R-CNN model, it can be effectively implemented in situations that demand real-time object detection. As illustrated in Figure 3 below, the architecture of the Faster R-CNN algorithm is designed to optimize this particular functionality.

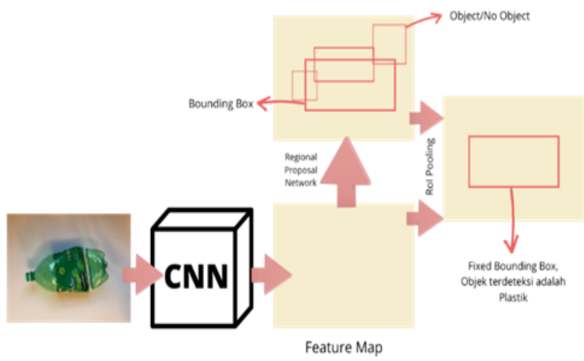

Figure 3. Faster R-CNN algorithm architecture

A more expedient R-CNN methodology has been developed to address the issue of sluggish image processing speed. This approach incorporates a Deep CNN for region proposal and a Fast R-CNN for the utilization of the proposed regions. The advantage of R-CNN over alternative models lies in its utilization of a selective search method. The region proposal network (RPN) predominantly informs the R-CNN as to the precise locations it should focus on [28].

The primary contrast between Fast Region-based

Convolutional Neural Network (R-CNN) and Faster R-CNN is the utilization of Selective Search in Fast R-CNN's logic, which strives to obtain the Region Proposal output. In contrast, the Faster R-CNN algorithm does not incorporate any external techniques, such as selective search. Instead, the Faster R-CNN utilizes the Regional Proposal Network (RPN) to accomplish this task. The RPN architecture can be seen in the Figure 4 below.

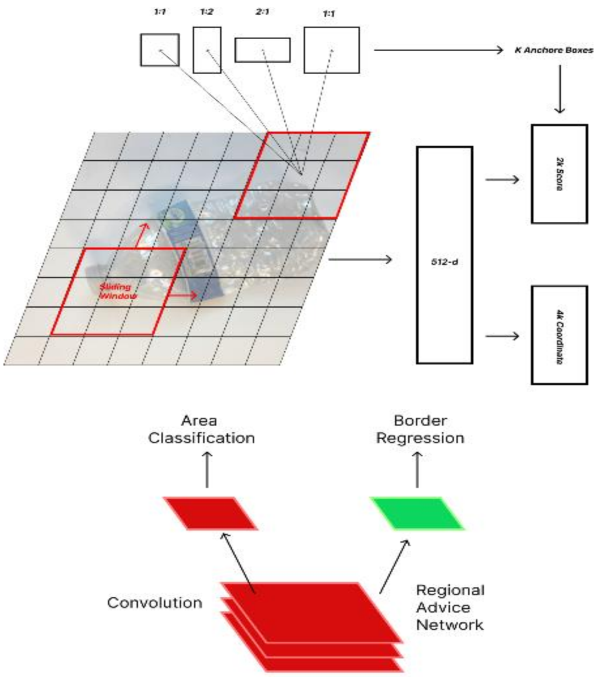

Figure 4. RPN architecture

3. RESULT AND DISCUSSION

3.1 Flutter

Flutter, a mobile application development framework, is commonly utilized for creating Android-based mobile applications. This particular framework is founded on the Dart programming language, which allows developers to efficiently create multiplatform applications with great ease and flexibility.

Dart offers an attractive mix of features, including strong typing, fast compilation, and a comprehensive standard library. Its strong typing system improves code reliability by enforcing strict type checks during the development process, reducing the potential for errors during the run. One of Dart's important advantages lies in its ability to compile code ahead of time, resulting in high-performance and optimized applications. This feature is particularly valuable in the context of web development, where the user experience relies heavily on fast load times and responsive interfaces. In addition, Dart comes with rich standard libraries that simplify common programming tasks, ranging from asynchronous programming support to web app frameworks such as Flutter, which is renowned for its ability to develop cross-platform mobile apps with a single code base.

One of the notable features that confer a distinct advantage to the Flutter framework is the Hot-Reload capability, which facilitates the effortless updating of the developed application without necessitating any recompilation. This feature thereby presents an expedited and streamlined development cycle, which is a highly sought-after attribute in modern software development paradigms. The Hot-reload functionality operates in a sophisticated manner by seamlessly integrating new code that has been meticulously crafted by the developer into the running Dart VM. This seamless integration mechanism effectively ensures that all the movements and actions of the application are meticulously preserved and meticulously conserved after the successful execution of the hot-reload operation [29].

The Flutter framework, a relatively new software tool utilized for mobile application development, was initially developed and continues to be primarily supported by Google, a renowned tech company. The first version of this innovative framework was launched in the year 2017, and since then it has garnered a significant amount of attention and popularity within the software development community. One of the standout features of Flutter, which sets it apart from other mobile programming languages, is its singular codebase, whereby the code for both logic and user interface are not bifurcated as is customary, but instead are integrated together.

Flutter, a mobile app development framework, essentially comprises various widgets that serve as the fundamental building blocks of all Flutter applications. It is essential to note that these widgets can be classified into two categories: stateful and stateless. The latter, more specifically, lack an internal state, which means that they cannot be modified after their creation. However, the former can be dynamically modified by altering their internal state without reinitialization. Stateful widgets are particularly indispensable when developing interactive applications in Flutter. To facilitate this, the Flutter framework creates states for its widgets, which can be read synchronously at the time of widget creation and can be modified throughout the lifecycle of the application [30]. Figure 5 below is the flutter architecture.

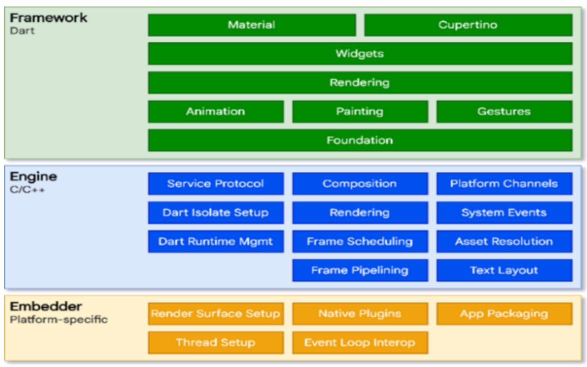

Figure 5. Flutter architecture

3.2 TensorFlow

In order to facilitate the expected functioning of the application, the employment of TensorFlow is deemed indispensable. TensorFlow is an end-to-end platform that has gained significant prominence in the machine-learning arena due to its remarkable ability to effectively train models. Once the model has been trained with TensorFlow, it presents the possibility of seamless implementation on an array of platforms including web, desktop, web, and cloud.

TensorFlow is an open-source software library that was initially created by Google for an internal project. It was later released to the public in late 2015, and the first version of TensorFlow was introduced in 2017. TensorFlow is a highly proficient tool that can cater to a large scale of training and inference requirements. It is capable of harnessing the power of numerous high-potency servers, some of which are GPU-enabled, that aid in the rapid training of models. Furthermore, it efficiently runs the trained models to facilitate production inference on various platforms. These platforms span from large-scale distributed clusters that are present in data centers to local execution on mobile devices. The tool is highly versatile and can support a diverse range of experimentation and research that pertains to the development of novel machine-learning models and system-level optimization [31].

3.3 Application workflow

The present Android application has been developed employing the Flutter framework, which is an open-source mobile application development platform that allows programmers to build high-performance, high-fidelity, apps for Android and iOS, from a single codebase. The main functionality of the aforementioned app comprises a button, which serves as a trigger to access the camera. Once the camera has been accessed, it captures an object, which is intended to be detected. After the object has been captured in the form of an image, the image is uploaded to the World Wide Web, where it is hosted using the Python programming language. It is worth noting that the TensorFlow model has been installed in the hosting server. Then, the image detection process will be carried out by the model that has been previously trained, using machine learning algorithms. The outcomes of the detection will be sent back to the Android application, where they will be displayed on the interface, in a visually appealing manner. For more details, see Figure 6.

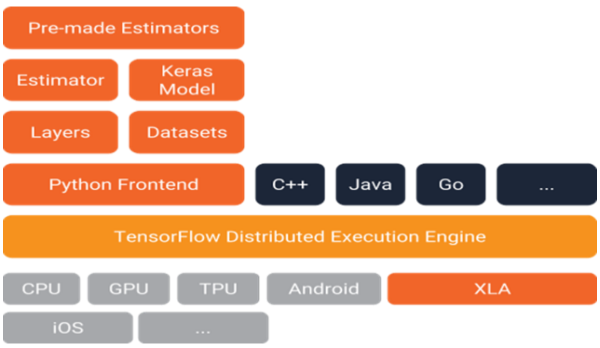

Figure 6. TensorFlow architecture

3.4 Design implementation

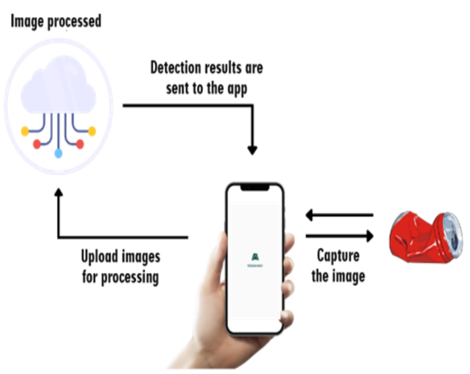

Figure 7. Application workflow

The implementation of the interface design is done using the Dart programming language. Interface implementation is done by following the design that has been made before. Below is the implementation of the design of the system that has been made shown in Figure 7.

3.5 System testing results

In order to conduct tests on the waste detection application pertaining to diverse waste categories of cardboard, glass, plastic, and paper, a pictorial approach is employed. Each type of waste is taken three times with each shooting session. In an effort to ensure that the machine learning model functions optimally, the object is rotated by approximately 20 degrees to obtain a range of accuracy values. This procedure is essential in validating the effectiveness of the machine learning model, thus contributing significantly to the field of waste management.

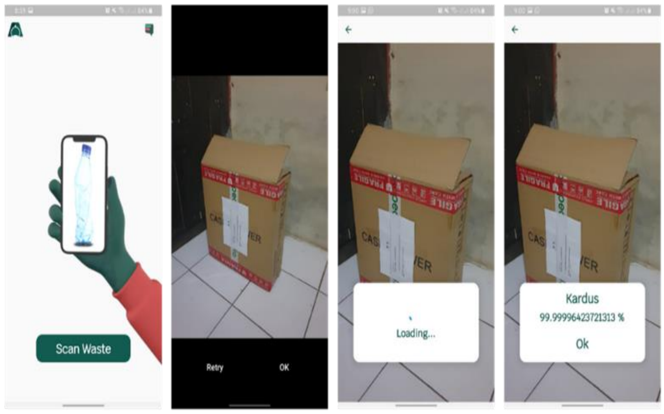

Figure 8. Interface view

The interface view can be observed in Figure 8, as depicted above. The results of object testing are used to evaluate the performance of the Android-based waste type detection model using the Faster R-CNN method. These results are analyzed to calculate the accuracy or level of accuracy of the model in detecting waste. This accuracy calculation is done using the following formula:

$$Accuracy = \frac{Total\ accuracy}{Total\ citra} \tag{1}$$

where, “total accuracy” represents the numerical value of the count of the correctly detected images, and “total image” denotes the aggregate number of images that have undergone testing in the model. Consequently, it can be inferred that a higher accuracy value is indicative of the superior performance of the waste type detection model that has been formulated. Thus, the outcomes derived from the exhaustive analysis of the system encompassing a comprehensive evaluation of as many as 12 images, as demonstrated in Table 1, are expounded upon in the ensuing discourse:

$$Accuracy = \frac{1177{,}273\%}{12} = 98{,}106\% \tag{2}$$

Based upon the computation of the precision coefficient indicated above, it has been inferred that the precision outcome is at a staggering 98.106% while taking into account a total of 12 data units for the complete image. This inquiry has demonstrated that the researcher's proposed method has outperformed various antecedent examinations in the realm of implementation assessment. In comparison to the research [32], it has been observed that the CNN method for identifying waste types has resulted in an accuracy rate of 87%. On the other hand, research [33] has used three pre-trained CNN models (VGG19, DenseNet169, and NAS Net Large) with waste type detection obtaining an accuracy rate of 94%. in the comparative analysis of these research studies, several key factors stand out as contributors to the differences in accuracy. Firstly, data collection methodologies varied across the studies, with one study employing a comprehensive approach involving multiple shots and object rotations, possibly leading to higher accuracy by capturing a broader range of waste scenarios. In contrast, another study lacked specific details about its dataset, and the exact data collection methods were not specified in two other studies, leaving room for variations in data quality and diversity. The diversity and quality of training data significantly impact a machine learning model's performance, and this variability in data collection approaches likely played a role in the differences observed. Secondly, the choice of model architecture and complexity appeared to influence accuracy significantly. These variations highlight that the selection of a suitable model architecture and complexity level can substantially affect the accuracy of waste classification systems.

Table 1. System test results

No	Object	Object Description	Trial Number	Output Label	Accuracy Result
1		Glass Jar	1	Glass	99.974%
			2	Glass	99.999%
			3	Glass	99.534%
2		Paper Brochure	1	Paper	99.999%
			2	Paper	99.998%
			3	Paper	99.970%
3		Plastic Bottle	1	Plastic	99.991%
			2	Plastic	98.419%
			3	Plastic	79.392%
4		Cardboard	1	Cardboard	100.0%
			2	Cardboard	99.999%
			3	Cardboard	99.998%
		Average Accuracy Result			**98.106%**

4. CONCLUSIONS

The test results show that the machine learning model using the Faster R-CNN algorithm achieved an average accuracy of 98.106% from twelve object captures. However, there are some limitations in this study, such as the slow detection process and reliance on the internet. This may affect the user experience and present challenges related to privacy and data consumption. To address these issues, several changes are proposed, such as storing data locally, improving data privacy and security, managing data consumption, and reducing dependency on external infrastructure. With these changes, it is expected that the application will become faster, more efficient, and more reliable, thus improving its performance and usability in real-world scenarios. As a suggestion for future researchers, it is recommended to integrate the machine learning model using the Faster R-CNN algorithm directly into the Android application itself, eliminating the need to upload objects to the web/cloud.

REFERENCES

[1] Xin, Y., Kong, L., Liu, Z., Chen, Y., Li, Y., Zhu, H., Gao, M.C., Hou, H.X., Wang, C. (2018). Machine learning and deep learning methods for cybersecurity. IEEE Access, 6: 35365-35381. https://doi.org/10.1109/ACCESS.2018.2836950

[2] Khosravy, M., Nakamura, K., Nitta, N., Babaguchi, N. (2020). Deep face recognizer privacy attack: Model inversion initialization by a deep generative adversarial data space discriminator. In 2020 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC), pp. 1400-1405. http://dx.doi.org/10.3837/tiis.2021.03.015

[3] Morra, L., Delsanto, S., Correale, L. (2019). Artificial intelligence in medical imaging: From theory to clinical practice. CRC Press. https://doi.org/10.1201/9780367229184

[4] Sahiner, B., Pezeshk, A., Hadjiiski, L.M., Wang, X., Drukker, K., Cha, K.H., Summers, R.M., Giger, M.L. (2019). Deep learning in medical imaging and radiation therapy. Medical Physics, 46(1): e1-e36. https://doi.org/10.1002/mp.13264

[5] LeCun, Y., Bengio, Y., Hinton, G. (2015). Deep learning. Nature, 521(7553): 436-444. http://doi.org/10.1038/nature14539

[6] Li, J., Zhang, D., Zhang, J., Zhang, J., Li, T., Xia, Y., Yan, Q., Xun, L. (2017). Facial expression recognition with faster R-CNN. Procedia Computer Science, 107: 135-140. https://doi.org/10.1016/j.procs.2017.03.069

[7] Al Amin, I.H., Arby, F.H., Winarno, E., Hartono, B., Hadikurniawati, W. (2022). Real-time social distance detection using YOLO-v5 with bird-eye view perspective to suppress the spread of COVID-19. In 2022 2nd International Conference on Information Technology and Education (ICIT&E), pp. 269-274. https://doi.org/10.1109/ICITE54466.2022.9759552

[8] Al Amin, I.H., Marinda, D.E., Winarno, E., UN, D.H., Lusiana, V. (2023). Real-time detection of face mask using convolutional neural network. Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi), 7(3): 697-704. http://doi.org/10.29207/resti.v7i3.5036

[9] Popel, M., Tomkova, M., Tomek, J., Kaiser, Ł., Uszkoreit, J., Bojar, O., Žabokrtský, Z. (2020). Transforming machine translation: A deep learning system reaches news translation quality comparable to human professionals. Nature Communications, 11(1): 4381. https://doi.org/10.1038/s41467-020-18073-9

[10] Kumar, A., Verma, S., Mangla, H. (2018). A survey of deep learning techniques in speech recognition. In 2018 international conference on advances in computing, communication control and networking (ICACCCN), pp. 179-185. http://doi.org/10.1109/ICACCCN.2018.8748399

[11] He, X., Yang, X., Zhang, S., Zhao, J., Zhang, Y., Xing, E., Xie, P. (2020). Sample-efficient deep learning for COVID-19 diagnosis based on CT scans. Medrxiv, 2020-24. https://doi.org/10.1016/j.neucom.2022.06.076

[12] Addae, G., Oduro-Kwarteng, S., Fei-Baffoe, B., Rockson, M.A.D., Antwi, E., Ribeiro, J.X.F. (2023). Patterns of waste collection: A time series model for market waste forecasting in the Kumasi Metropolis, Ghana. Cleaner Waste Systems, 4: 100086. https://doi.org/10.1016/j.clwas.2023.100086

[13] Ribeiro-Rodrigues, E., Bortoleto, A.P., Fracalanza, B.C. (2021). Exploring the influence of contextual and sociodemographic factors on waste prevention behaviour-the case of Campinas, Brazil. Waste Management, 135: 208-219. http://doi.org/10.1016/j.wasman.2021.09.002

[14] Johnson, N.E., Ianiuk, O., Cazap, D., Liu, L., Starobin, D., Dobler, G., Ghandehari, M. (2017). Patterns of waste generation: A gradient boosting model for short-term waste prediction in New York City. Waste Management, 62: 3-11. http://doi.org/10.1016/j.wasman.2017.01.037

[15] Ferdous, W., Manalo, A., Siddique, R., Mendis, P., Zhuge, Y., Wong, H.S., Lokuge, W.P., Aravinthan, T., Schubel, P. (2021). Recycling of landfill wastes (tyres, plastics and glass) in construction–A review on global waste generation, performance, application and future opportunities. Resources, Conservation and Recycling, 173: 105745. https://doi.org/10.1016/j.resconrec.2021.105745

[16] Pan, A., Yu, L., Yang, Q. (2019). Characteristics and forecasting of municipal solid waste generation in China. Sustainability, 11(5): 1433. https://doi.org/10.3390/su11051433

[17] Amo-Asamoah, E., Owusu-Manu, D.G., Asumadu, G., Ghansah, F.A., Edwards, D.J. (2020). Potential for waste to energy generation of municipal solid waste (MSW) in the Kumasi metropolis of Ghana. International Journal of Energy Sector Management, 14(6): 1315-1331. https://doi.org/10.1108/IJESM-12-2019-0005

[18] Girshick, R., Donahue, J., Darrell, T., Malik, J. (2014). Rich feature hierarchies for accurate object detection and semantic segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 580-587. http://dx.doi.org/10.1109/CVPR.2014.81

[19] Girshick, R. (2015). Fast R-CNN. In Proceedings of the IEEE International Conference on Computer Vision, pp. 1440-1448. https://doi.org/10.1109/ICCV.2015.169

[20] Zhou, T., Li, Z., Zhang, C. (2019). Enhance the recognition ability to occlusions and small objects with robust faster R-CNN. International Journal of Machine Learning and Cybernetics, 10: 3155-3166. https://doi.org/10.1007/s13042-019-01006-4

[21] Liu, X., Wang, H., Jing, H., Shao, A., Wang, L. (2020). Research on intelligent identification of rock types based on faster R-CNN method. IEEE Access, 8: 21804-21812. http://dx.doi.org/10.1109/ACCESS.2020.2968515

[22] Shibly, K.H., Dey, S.K., Islam, M.T.U., Rahman, M.M. (2020). COVID faster R–CNN: A novel framework to Diagnose Novel Coronavirus Disease (COVID-19) in X-Ray images. Informatics in Medicine Unlocked, 20: 100405. https://doi.org/10.1016/j.imu.2020.100405

[23] Nguyen, C.C., Tran, G.S., Burie, J.C., Nghiem, T.P. (2021). Pulmonary nodule detection based on faster R-CNN with adaptive anchor box. IEEE Access, 9: 154740-154751. https://doi.org/10.1016/j.bea.2023.100076

[24] Zhang, J., Wang, X., Ni, G., Liu, J., Hao, R., Liu, L., Liu, Y., Du. X., Xu, F. (2021). Fast and accurate automated recognition of the dominant cells from fecal images based on Faster R-CNN. Scientific Reports, 11(1): 10361. https://www.nature.com/articles/s41598-021-89863-4

[25] Xu, B., Wang, W., Falzon, G., Kwan, P., Guo, L., Chen, G., Tait, A., Schneider, D. (2020). Automated cattle counting using Mask R-CNN in quadcopter vision system. Computers and Electronics in Agriculture, 171: 105300. https://doi.org/10.32604/csse.2023.034192

[26] Ren, S., He, K., Girshick, R., Sun, J. (2015). Faster R-CNN: Towards real-time object detection with region proposal networks. Advances in Neural Information Processing Systems, 28. http://dx.doi.org/10.1109/TPAMI.2016.2577031

[27] Uijlings, J.R., Van De Sande, K.E., Gevers, T., Smeulders, A.W. (2013). Selective search for object recognition. International Journal of Computer Vision, 104: 154-171. http://dx.doi.org/10.1007/s11263-013-0620-5

[28] Mohammed, H., Tannouche, A., Ounejjar, Y. (2022). Weed detection in pea cultivation with the faster RCNN ResNet 50 convolutional neural network. Revue d'Intelligence Artificielle, 36(1): 13-18. https://doi.org/10.18280/ria.360102

[29] Wu, W. (2018). React native vs flutter, cross-platforms mobile application frameworks. https://urn.fi/URN:NBN:fi:amk-201805158156

[30] Boukhary, S., Colmenares, E. (2019). A clean approach to flutter development through the flutter clean architecture package. In 2019 International Conference on Computational Science and Computational Intelligence (CSCI), pp. 1115-1120. https://doi.org/10.1109/CSCI49370.2019.00211

[31] Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., et al. (2016). TensorFlow: A system for Large-Scale machine learning. In 12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16), pp. 265-283. https://doi.org/10.48550/arXiv.1605.08695

[32] Adedeji, O., Wang, Z. (2019). Intelligent waste classification system using deep learning convolutional neural network. Procedia Manufacturing, 35: 607-612. https://doi.org/10.1016/j.promfg.2019.05.086

[33] Huang, G.L., He, J., Xu, Z., Huang, G. (2020). A combination model based on transfer learning for waste classification. Concurrency and Computation: Practice and Experience, 32(19): e5751. https://doi.org/10.1007/s11042-020-09571-5