Enhancing Object Tracking in Augmented Reality Using Convolutional Neural Network-Based

Nurhadi[1], Deris Stiawan[2], Mohd. Yazid Idris[3], Saparudin[4]
[1]Department of Informatics Engineering, Dinamika Bangsa University; Faculty of Engineering, Sriwijaya University, Indonesia
[2]Department of Computer Engineering, Universitas Sriwijaya, Indonesia
[3]School of Computing, Faculty of Engineering, Universiti Teknologi Malaysia
[4]Department of Informatics Engineering, Telkom University, Indonesia

Article Info

Article history:

Received Aug 26, 2022
Revised Nov 9, 2022
Accepted Nov 28, 2022

Keywords:

Augmented reality
Deep learning
Object tracking
Markerless
CNN

ABSTRACT

Augmented reality (AR) has been applied in maintenance, simulation, remote assistance, and other fields. One of the issues arising from these applications is how objects can be placed in physical environments using an AR system. Object placement consists of two processes: object detection and segmentation. Due to the importance of placement, in this paper, we propose using deep learning to address issues with the placement of objects through detection and segmentation in AR. Deep learning can help complete tasks by providing correct information about environmental changes in real-time situations. The problem is that it is rarely used in AR, which suggests a combination of deep learning-based object detection and instance segmentation with wearable AR technology to improve performance on complex tasks. In our work, we propose to address this problem by applying a convolutional neural network to facilitate the detection and segmentation of objects in real environments. To measure AR performance, we examined detection accuracy in environments with different intensities. The results of the experiment demonstrate satisfactory performance, reaching 98% for segmentation and accurate detection.

Corresponding Author:

Nurhadi,
Department of Informatics Engineering,
Dinamika Bangsa University & Faculty of Engineering Sriwijaya University,
Palembang, Indonesia.
Email: nurhadi@unama.ac.id

1. INTRODUCTION

In augmented reality (AR), two types of techniques are used to perform augmentation: marker-based and markerless. In the context of AR, marker-based techniques refer to what are commonly known as fiduciary markers. Meanwhile, markerless techniques are used in applications that do not require an anchor to the real world or a physical (fiduciary) marker. The display of augmented objects not only shows them as floating but also allows 3D DL objects to be automatically placed onto a flat surface to increase realism and real-time responses [1].

Most AR technologies track image markers and use them to add virtual objects. This is fast and accurate. However, AR markers must be registered first; they must be attached to real objects, and the positions of virtual objects relative to physical objects must be determined. Thus, it is difficult to support AR without fiducial markers [2]. In addition, instability and mismatch issues often occur when the object is far from the AR camera or AR marker [3].

Machine learning has been applied to solve difficult problems, such as product defect detection, facility preventive maintenance, and object detection. Deep learning (DL) is a subarea of machine learning based on the artificial learning of neural networks consisting of many deep, complex layers [4]. Representative DL methods include convolutional neural networks (CNNs) with good image classification performance, such as visual object detection and human pose estimation, and repetitive neural networks, which are used for forms of sequential data analysis, such as speech recognition and machine translation. In addition, the two types of networks can be combined in texts and visual images [5].

Although physical objects can be detected using DL methods, it is important to provide an easy-to-understand form for visualizing information. Therefore, AR technology has been widely utilized to provide realistic three-dimensional (3D) results for visualization and interaction in various fields, such as maintenance, training, interaction, and assembly applications [5-7]. Thus, area learning-based AR enables the augmentation of 3D virtual objects and supports more effective interactions with them without AR markers [8]. However, it is difficult to detect object information, such as the class, position, and pose of a real object, by scanning the surface of a physical object using 3D or by identifying key features.

In this study, an intelligent task assistance approach is proposed in which DL-based object detection and instant segmentation are combined with wearable AR technology to achieve better performance on complex tasks. The use of DL can help users complete tasks by providing the correct information effectively while taking into account the dynamically changing environment and the user's situation in real time. In particular, CNN is used to efficiently perform object detection and instant segmentation of physical objects [7], [9]. The results of instance segmentation are displayed visually on the AR display. Object detection, instance segmentation, and wearable AR combine to find reconstructed 3D maps and regions with actual objects in the physical environment.

The contributions of this study are (1) proposing an architecture for the detection and segmentation of physical objects in AR using CNN and (2) measuring the performance of the proposed method with different environmental intensities. The remainder of this paper is organized as follows: Section 2 presents the relevant research. Section 3 briefly discusses the experimental setup used in this study. Section 4 describes the experiment and findings of this study in more detail. Finally, Section 5 provides the conclusions and discusses potential future work.

2. RELATED WORK

The use of DL in AR is still relatively new. Several previous studies have proposed the use of AR, such as [8], [10] (DNN), [11] (LSTM), and [12-15] (CNN). One study [16] used a deep neural network (DNN) for an intelligent municipality AR service system in the fields of information dissemination and tourism. The results of this study indicate a higher detection accuracy (67.5%) compared to naïve Bayes and random forest. Another study [17] also proposed a DNN method for a clinical decision support system (CDSS) used to classify bedsore disease. The results of the CDSS test showed that the accuracy was quite satisfactory, reaching 89%.

Another DL method proposed for AR systems is LSTM, which was applied to assist predictive maintenance in internet of things (IoT) manufacturing [11]. This study compared LSTM and CNN methods to improve the performance of AR in maintenance in a manufacturing environment. In addition, [12] proposed lightweight DL on mobile-based using CNN. Source [12] is an initial study that has just been published, so the results are not yet satisfactory. The use of CNN methods has also been proposed by several previous studies [15-17] to improve the performance of AR for system maintenance. Based on these studies, the performance of AR is quite satisfactory. Therefore, in this study, we propose the use of CNN to aid in the detection process and object segmentation in lab maintenance (Arduino). The contribution of this study is to improve the performance of the CNN method previously proposed in [12]. Study [12] only shows the initial state of the research, so there is a need for further research using CNN in the AR field for detection and segmentation. Therefore, this research proposes a CNN method to improve the performance of AR object detection and segmentation with different light intensities.

Table 1. Previous research

Ref.	Method	Dataset/Environment	Measurement
[16]	DNN	Premier dataset (Jeddah)	Detection rate
[17]	DNN	Bedsore dataset	Precision, Recall, F-score, Accuracy
[11]	LSTM	Tools used in IoT for manufacturing	Running time, MAE, RMSE, Accuracy
[18]	DFFNN	Sensor aquaculture prawn farm	Accuracy
[9][10][11][15]	CNN, CNN+RGB, R-CNN	COCO dataset	MAE, RMSE, Running time
This study	CNN	COCO dataset (pre-training) and physical environment lab (Arduino)	Accuracy, Intensity

3. RESEARCH METHOD

This section describes the experimental setup, DL, and architecture of the proposed method and the experimental configuration. This study proposes a DL method based on a CNN to improve detection and segmentation processes in AR systems. This experiment has several stages: an AR interface, DL, visualization augmentation, and task assistance.

3.1. Dataset

In this study, the COCO dataset was used for DL training for the detection and segmentation processes in a real environment. The COCO dataset is a collection of image datasets consisting of training and testing images. In addition, there are also annotated data from the point of object detection results. This dataset has been used in previous studies [9], [15]. An Arduino device was used in the detection of physical objects in this study. The COCO dataset had 78,458 images in 2017, but five classes of 10 images were used for pre-training. In the testing, Arduino devices were used for experiments on the use of physical object data. The following figure presents sample images from the COCO dataset and a physical object.

Figure 1. A few instances of COCO dataset images

3.2. Experimental Setup

In general, there are four main stages of the experimental setup, as shown in Figure 2: (1) an interface AR module, (2) a deep learning module, (3) a visual augmentation module, and a task assistance module. These can be described as follows:

- Interface AR module – The AR interface module enables AR devices to capture images for physical object recognition and deliver them to the DL module.
- DL module – The DL module detects physical objects in captured images and then implements the segmentation of objects before sending the findings to the AR interface module. It operates on a remote server and can handle numerous requests from AR or other smart devices at the same time.
- Visual augmentation module – Visual augmentation scans a real environment utilizing AR devices and assembles a 3D map when the user moves around in the real environment. At the same time, the visual augmentation matches the places of real objects in the 3D map as soon as the DL module identifies real objects. This study used a Unity 3D engine to build an AR environment for AR devices.
- Task assistance module – Finally, the task assistance module enables users to execute required tasks more effectively, including assembly maintenance and operation.

3.3. Proposed Method for AR Using a Convolutional Neural Network

DL is an advanced subfield of machine learning that brings machine learning closer to artificial intelligence. It enables the modeling of complicated relationships and concepts by utilizing many layers of representation. Supervised and unsupervised learning algorithms are used to produce increasingly higher levels of abstraction defined by the output properties of previous levels. Various methods are used in DL, such as deep belief networks (DBNs) [19], autoencoders [20], recurrent neural networks (RNNs) [21], long short-term memory networks (LSTMs) [22], and convolutional neural networks (CNNs) [13][23-24]. This study proposes the use of a CNN for detection and segmentation methods in AR. One of the reasons for using a CNN in this study is that several previous studies have produced satisfactory performance using this method with AR systems.

CNN is a DL technique popularly used for object detection. A picture in a CNN must be split into pieces. The CNN takes each of these elements as input and produces the object class after passing them through convolution and pooling layers [25]. Figure 2 depicts how the CNN is used in DL to recognize actual objects and separate their areas in images acquired by AR after training the dataset. The learning parameters for object recognition and instance segmentation implemented in the user research employing CNN are shown in Table 2. TensorFlow was used for training and testing. The COCO dataset was used to develop a pretrained CNN model, and ResNet structures were employed. We trained 10 classes for the first step of user research and 30 classes for the second. Training was carried out using a stochastic gradient descent optimizer, and 0.0001 was the learning rate. The CNN's structure was unusual. The convolution layer consisted of one stack on top of another, followed by several layers, and the end was a fully connected layer. The convolution layer was the most important part of the CNN, as it extracted the local model and kernel weight-sharing mechanism. The CNN in this study consisted of two parameters—the trained weight "W" and bias "b"—that were used to generate an output feature model with, e.g., a point of W:

$$fi = A\,(Wi \times xi - 1 + bi) \tag{1}$$

where fi is the initial feature of the i-th convolution layer, A is the activation function, Wi and bi are the values for the weight and bias in the i-th convolution layer, and xi−1 is the output feature model of the last pooling layer. The pooling layer is a part of the CNN method used to intermediate a layer between two convolution layers. This part was used to increase efficiency and avoid overfitting during training procedures. The MaxPooling method was used for the pooling function. It reduced the sample from the input feature using a non-linear max function to derive the number of convolutional layer parameters. The feature extraction process at the pooling layer can be seen below.

$$Xj = MaxPool(fj) \tag{2}$$

$$yk = Wk\ x\ Yk - 1 + bk \tag{3}$$

The full layer of the CNN can be defined as yk, where yk is the k-th full layer output vector, and Yk−1 is the output in the k-th layer.

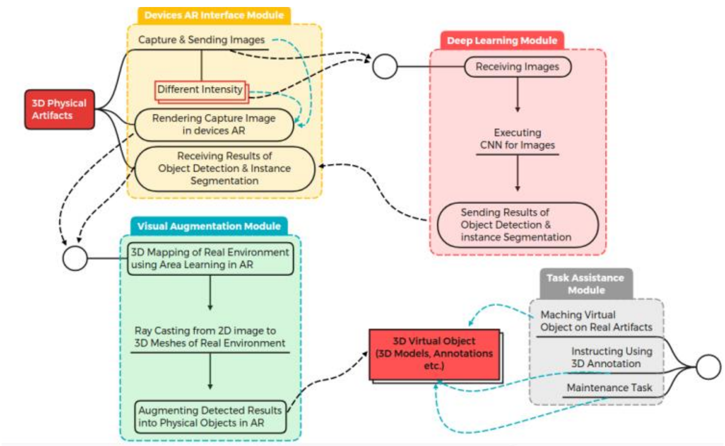


Figure 2. Design of the proposed architecture

Pseudocode 1: The CNN Phase

1. **Input**: d dataset, 1 dataset label, W Vector matrix
2. **Output**: result of CNN model
3. **Let** f be feature set 3D
4. **For** a in dataset **do**
5. Let f_a be the feature matrix of sample a
6. **For** b in a **do**
7. V_b ← vectorise ($_{b,\,w}$)
8. **Append** V_b to f_a
9. **Append** f_a to f
10. **End for**
11. **End for**
12. Ftrain, Ftest, Itrain, Itest ←**split** feature set and labels
13. X ←CNN (Ftrain, Ltrain)
14. Score ←evaluate (a, Itest, M)
15. **Return** Score

Table 2. Parameter environment

Parameters	Descriptions
Environment	TensorFlow
Architecture	CNN
Dataset	5(10) Classes
Pre-train dataset	COCO dataset
Epoch	100
Learning rate	0.0001

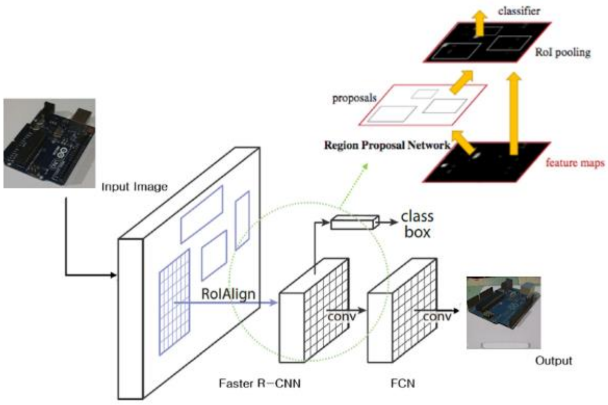


Figure 2. CNN architecture used in this study

A detailed explanation of the object detection process in AR using the CNN method is shown in **Pseudocode 1**. The training process is divided into two inputs: the COCO dataset as pre-training input and the physical environment. The environmental parameters for the training process are shown in Table 2. Certain stages must be passed before the detection process, including the segmentation process. **Pseudocode 2** provides a detailed explanation of the segmentation process in the method proposed in this study.

Pseudocode 2: Segmentation Phase

```
Input: d COCO dataset
Output: Segmented images
Training data augmentation
 For I in data do
  Initialize numpy → size (x, y, z)
   Returns ←(augmented image)
  Returns ←(corresp, mask(imagei))
 End for
 PacthSize ←(array*array, shape)
Augmented ← transform()
AugmentedMask ← transforms(spatial)
Mask ← mask[0]
Return (augmented[0], mask)
 For each I in d
 Returns ←sample(weights)
nSample ← length(selflab)
cOuntDic ← dict(unique, count)
weights ← [ ]
For a label in selflab do
  Append (weights) ← nSample/cOuntDic  label
  Return weights
  End for
End for
Update weights
While (iter<=imagei)
End
```

3.4. Experimental Configuration

This work used a computer with an Intel Core i7 processor, 2.60 GHz, and 12 GB of RAM. The operating system was Ubuntu 20.04.3 LTS. The markerless AR design in this work was built using Unity and Vuforia DL. The experiments used Android for markerless AR testing. Finally, for the 3D dataset, the object used was the Arduino model.

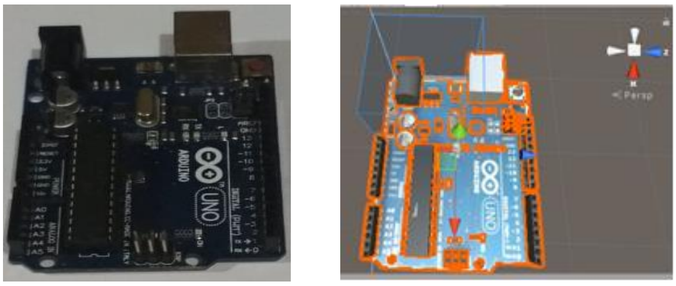
4. RESULTS AND DISCUSSION

This section presents the system preparation, the experimental details of using CNN for object detection and segmentation, a comparison of light intensities, and an experimental discussion. The COCO dataset was used to test the proposed method with five classes of 10 images for pretraining. The Arduino functioned as a real object that was entered into the AR system using the feature extraction method, namely

CNN. The measurements in this work were meant to determine the accuracy and the effects of light intensity on the detection and segmentation process and to display the AR system. The scanner environment was set up using the following specifications: a light intensity from 50 cd–200 cd, a camera lens of more than 8 MP, a distance of the object from the camera of 25 cm, and the scanning view in the top, left-right.

4.1. Object Tracking using a CNN

This stage presents the results of the object scanning and detection processes intended to extract features from the target object. Figure 3a shows the real target from which features were extracted, and Figure 3b shows the results of the feature extraction using the CNN. The result of the feature extraction was a point from every corner of the target object. The next process implemented was ray casting to cast objects that were less than perfect. The feature extraction was highly dependent on the environmental lighting and camera lens quality.



(a) (b)

Figure 3. Scanning an object using DL: (**a**) real objects; (**b**) results of scanning object features using DL.

4.2. Results of the Proposed Method

The proposed method was evaluated in several scenarios to measure the performance of the AR system. The first was to measure the accuracy or success of the CNN method in feature extraction and segmentation. The next step was to measure the effects of the environment on object detection and segmentation when exposing objects to the AR frame.

Table 3 shows the results of experiments using CNN to detect and segment objects. The experiment was carried out with several object conditions, such as the Arduino. Real objects were set in different positions for each test. The position of the extracted object shifted from the horizontal and diagonal positions to the vertical position relative to the position of the angle lens. The experimental results showed that the CNN was successful in extracting features from real objects and displaying them on the AR screen.

Table 3. Example of AR using DL

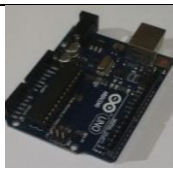
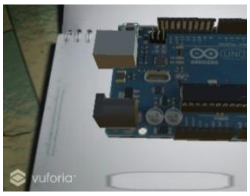
Real environment	3D	Real environment	3D
			
			
			

Table 4 shows the results with different intensity inputs. Different intensities produced significantly different results. In essence, the intensity is the numerical value of pixels extracted to obtain features of the target object. This work was carried out in five different scenarios with intensities > 50–200 cd. The results showed that at an intensity < 50, building augmentation objects in the AR system was not successful. The best results were obtained from experiments with intensities of more than 50–190 cd. Furthermore, at intensities > 200 cd, the object was not successfully exposed in the AR frame because the intensity was too high.

Table 4. Results of experiments with different intensities

Intensity	Results	Intensity	Results
< 50		80–110	
50–70		120–170	
180–190		> 200	Image could not be created

Subsequently, we carried out experiments to evaluate the performance of the proposed system using different intensity inputs. Ten experiments were conducted, with each experiment using 10 real targets and differences in intensity. Figure 4 shows the results of comparing the accuracy of the experiments that were carried out. The experimental results showed that the performance was not good at intensities < 50 and > 200, which were not successful for augmentation because they were too dark or too bright. The best results were obtained from experiments with intensities of 50–190 cd.

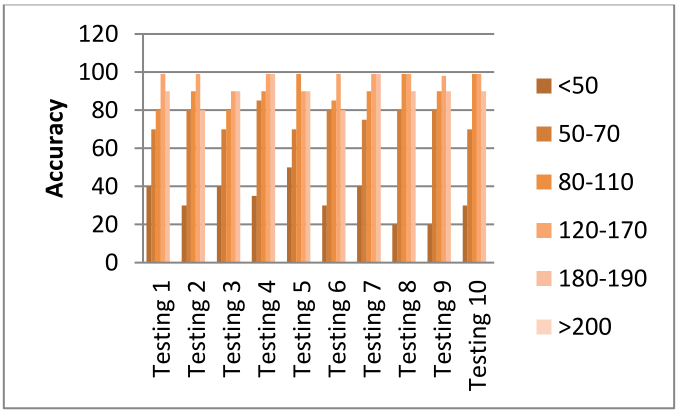


Figure 4. Comparison of the accuracies of different tests

4.3. Comparison with Previous Research

This section presents the results of the experiments that were carried out and compares them with the results of other studies. The aim was to compare the performance of the CNN with those of other methods in AR. However, some previous studies used different datasets. This is because AR systems use real testing data, so it is difficult to make comparisons using the same dataset and features. The result of this research is to improve the performance of AR using CNN. Table 5 is a comparison of the accuracy of the results reached using DL in this research. The experimental results show that the proposed method produced more accurate performance than those used in other studies. Therefore, it can be concluded that the proposed method is successful and exhibits satisfactory performance. However, some of the studies did not mention specific degrees of accuracy but merely claimed that the proposed methods had satisfactory accuracy.

Table 5. Comparison with previous research

Study	Method	Accuracy/Score (%)	Successful Object Recognition
[26]	Machine Learning	82.19	√
[18]	DNN	89.20	√
[9]	CNN-RGB-D	n/	√
[7]	CNN	n/	√
This study	CNN	98.98	√

.

5. CONCLUSION

In this paper, we described a DL-based method for detecting objects and segmenting instances in real-world situations. The suggested technique makes use of DL (specifically, a CNN) to improve the identification of real-world items in AR. To measure the usability of the proposed technique, we conducted two user studies that were used to evaluate practical tasks, such as object matching and the simulation of different intensities in a real environment. We contend that the suggested technique recognizes objects and segments instances successfully in an AR environment. However, some difficulties with the proposed approach must be addressed. Because the images were hazy when captured, the images could not identify the items properly. Moreover, due to the limited distance of the AR camera module, it was necessary to move closer to generate 3D meshes of actual spaces when detecting items at a great distance. In future research, we propose to use more varied physical object tests in real time. In addition, we intend to implement an AR system for more complex assistant systems.

ACKNOWLEDGMENTS

The University of Dinamika Bangsa Jambi provided funding for this research through a human resource development initiative.

REFERENCES

[1] Nurhadi, Saparudin, N. Adam, D. Purnamasari, Fachruddin, and A. Ibrahim, “Implementation of Object Tracking Augmented Reality Markerless using FAST Corner Detection on User Defined-Extended Target Tracking in Multivarious Intensities,” J. Phys. Conf. Ser., vol. 1201, no. 1, 2019, doi: 10.1088/1742-6596/1201/1/012041.

[2] J. Murithi and C. Lin, “Markerless cooperative augmented reality-based smart manufacturing double-check system : Case of safe PCBA inspection following automatic optical inspection,” Robot. Comput. Integr. Manuf., vol. 64, no. February, p. 101957, 2020, doi: 10.1016/j.rcim.2020.101957.

[3] S. Boonbrahm, P. Boonbrahm, and C. Kaewrat, “The use of marker-based augmented reality in space measurement,” Procedia Manuf., vol. 42, no. 2019, pp. 337–343, 2020, doi: 10.1016/j.promfg.2020.02.081.

[4] J. Schmidhuber, “Deep Learning in neural networks: An overview,” Neural Networks, vol. 61, pp. 85–117, 2015, doi: 10.1016/j.neunet.2014.09.003.

[5] H. Subakti and J. R. Jiang, “Indoor Augmented Reality Using Deep Learning for Industry 4.0 Smart Factories,” Proc. - Int. Comput. Softw. Appl. Conf., vol. 2, pp. 63–68, 2018, doi: 10.1109/COMPSAC.2018.10204.

[6] M. S. H. Sassi and L. C. Fourati, "Deep Learning and Augmented Reality for IoT-based Air Quality Monitoring and Prediction System," 2021 International Symposium on Networks, Computers and Communications (ISNCC), 2021, pp. 1-6, doi: 10.1109/ISNCC52172.2021.9615639.

[7] K. Park, M. Kim, S. H. Choi, and J. Y. Lee, “Deep learning-based smart task assistance in wearable augmented reality,” Robot. Comput. Integr. Manuf., vol. 63, no. October 2019, p. 101887, 2020, doi: 10.1016/j.rcim.2019.101887.

[8] J. -F. Lalonde, "Deep Learning for Augmented Reality," 2018 17th Workshop on Information Optics (WIO), 2018, pp. 1-3, doi: 10.1109/WIO.2018.8643463.

[9] K. Park, S. Ho, M. Kim, and J. Yeol, “Deep learning-based mobile augmented reality for task assistance using 3D spatial mapping and snapshot-based RGB-D data,” Comput. Ind. Eng., vol. 146, no. April, p. 106585, 2020, doi: 10.1016/j.cie.2020.106585.

[10] Y. Ghasemi, H. Jeong, S. H. Choi, K. Park, and J. Y. Lee, “Deep learning-based object detection in augmented reality: A systematic review,” Comput. Ind., vol. 139, p. 103661, 2022, doi: 10.1016/j.compind.2022.103661.

[11] C. Liu, H. Zhu, D. Tang, Q. Nie, T. Zhou, and L. Wang, “Robotics and Computer-Integrated Manufacturing Probing an intelligent predictive maintenance approach with deep learning and augmented reality for machine tools in IoT-

enabled manufacturing," Robot. Comput. Integr. Manuf., vol. 77, no. November 2021, p. 102357, 2022, doi: 10.1016/j.rcim.2022.102357.

[12] J. G. Ko, S. Lee, S. Lee and J. Lee, "Lightweight Deep Learning based Intelligent Mobile Augmented Reality," 2021 IEEE International Conference on Consumer Electronics-Asia (ICCE-Asia), 2021, pp. 1-3, doi: 10.1109/ICCE-Asia53811.2021.9641972.

[13] G. Lampropoulos, E. Keramopoulos, and K. Diamantaras, "Visual Informatics Enhancing the functionality of augmented reality using deep learning , semantic web and knowledge graphs : A review," Vis. Informatics, vol. 4, no. 1, pp. 32–42, 2021, doi: 10.1016/j.visinf.2020.01.001.

[14] V. Kohli, U. Tripathi, V. Chamola, B. Kumar, and S. S. Kanhere, "Microprocessors and Microsystems A review on Virtual Reality and Augmented Reality use-cases of Brain Computer Interface based applications for smart cities," Microprocess. Microsyst., vol. 88, no. December 2021, p. 104392, 2022, doi: 10.1016/j.micpro.2021.104392.

[15] Q. Duan et al., "Augmented Reality and Deep Learning Guided Task Oriented Robot," 2021 33rd Chinese Control and Decision Conference (CCDC), 2021, pp. 3241-3245, doi: 10.1109/CCDC52312.2021.9602607.

[16] A. Besse et al., "ScienceDirect iMars : Intelligent Municipality Reality Service for Efficient Information Dissemination based on Deep Learning iMars : Intelligent Municipality Augmented Reality Service for algorithm in Smart City of Jeddah Efficient Information Disseminat," Procedia Comput. Sci., vol. 163, pp. 93–108, 2019, doi: 10.1016/j.procs.2019.12.091.

[17] F. Orciuoli, F. J. Orciuoli, and A. Peduto, "ScienceDirect A Mobile Mobile Clinical Clinical DSS DSS based based on on Augmented Augmented Reality Reality and and Deep Deep Learning for the home cares of patients a ffl icted by bedsores Learning for the home cares of patients a ffl icted by bedsores," Procedia Comput. Sci., vol. 175, no. 2019, pp. 181–188, 2020, doi: 10.1016/j.procs.2020.07.028.

[18] A. Rahman et al., "Aquacultural Engineering An integrated framework of sensing , machine learning , and augmented reality for aquaculture prawn farm management," Aquac. Eng., vol. 95, no. August, p. 102192, 2021, doi: 10.1016/j.aquaeng.2021.102192.

[19] S. Sharipuddin et al., "Intrusion detection with deep learning on internet of things heterogeneous network," IAES Int. J. Artif. Intell., vol. 10, no. 3, p. 735, 2021, doi: 10.11591/ijai.v10.i3.pp735-742.

[20] M. Haggag, M. M. Tantawy, and M. M. S. El-Soudani, "Implementing a Deep Learning Model for Intrusion Detection on Apache Spark Platform," IEEE Access, vol. 8, no. Dl, pp. 163660–163672, 2020, doi: 10.1109/access.2020.3019931.

[21] S. Abirami and P. Chitra, Energy-efficient edge based real-time healthcare support system, 1st ed. Elsevier Inc., 2019.

[22] L. Wu and T. Deng, "Computer Network Security Analysis Modeling Based on Spatio-temporal Characteristics and Deep Learning Algorithm," J. Phys. Conf. Ser., vol. 1648, no. 4, 2020, doi: 10.1088/1742-6596/1648/4/042111.

[23] H. El Merabet and A. Hajraoui, "A Survey of Malware Detection Techniques based on Machine Learning," Int. J. Adv. Comput. Sci. Appl., vol. 10, no. 1, pp. 366–373, 2019, doi: 10.14569/ijacsa.2019.0100148.

[24] W. Wang, M. Zhao, and J. Wang, "Effective android malware detection with a hybrid model based on deep autoencoder and convolutional neural network," J. Ambient Intell. Humaniz. Comput., vol. 0, no. 0, pp. 1–9, 2018, doi: 10.1007/s12652-018-0803-6.

[25] A. Kim, M. Park, and D. H. Lee, "AI-IDS: Application of Deep Learning to Real-Time Web Intrusion Detection," IEEE Access, vol. 8, pp. 70245–70261, 2020, doi: 10.1109/ACCESS.2020.2986882.

[26] D. Conference, P. Stief, J. Dantan, A. Etienne, and A. Siadat, "Operator training framework for hybrid environments : An Augmented Reality module machine learning object recognition," Procedia CIRP, vol. 106, pp. 102–107, 2022, doi: 10.1016/j.procir.2022.02.162.

[27] T. K. Behera, P. K. Sa, M. Nappi, and S. Bakshi, "Satellite IoT Based Road Extraction from VHR Images Through Superpixel-CNN Architecture," *Big Data Res.*, vol. 30, p. 100334, 2022, doi: https://doi.org/10.1016/j.bdr.2022.100334.