*Journal of Advanced Research Design* 139, Issue 1 (2026) 144-157

Penerbit Akademia Baru

Journal of Advanced Research Design

Journal homepage: https://akademiabaru.com/submit/index.php/ard
ISSN: 2289-7984

# Automatic Wheelchairs Based on Artificial Intelligence for Disable People

Asep Sholahuddin[1,*], Anton Satria Prabuwono[2], Ino Suryana[1], Akik Hidayat[1]

[1] Department of Computer Science, Faculty of Mathematics and Natural Sciences, Universitas Padjadjaran, Sumedang, Jawa Barat 45363, Indonesia
[2] Faculty of Computing and Information Technology in Rabigh, King Abdulaziz University, Jeddah 22254, Saudi Arabia

**ARTICLE INFO**

***Article history:***
Received 28 February 2025
Received in revised form 28 March 2025
Accepted 14 July 2025
Available online 25 July 2025

***Keywords:***
Disable people; artificial intelligence; wheelchair; YOLOv8

**ABSTRACT**

Disabled people have difficulty in carrying out activities. For example, someone who can only work with his head and cannot move or track his hands and feet carries out all activities using his head. Robot technology has developed rapidly and can be applied to wheelchairs to help people with this kind of disability. Research on wheelchairs that can move using the head showed the problem that lengthy activities may cause the head to hurt. Therefore, wheelchairs that can track automatically from one room to another are needed. This research focuses on a wheelchair that could move automatically through the rooms in a large house automatically by recognizing room symbols that had been given and trained. The wheelchair was equipped with an Artificial Intelligence base and programmed using YOLOv8. The program could recognize 10 room symbols. The symbol recognition accuracy result was 100% for confidence 0.996.

## 1. Introduction

The wheelchair is essential for persons with disabilities of all age groups. Around 1.825 million wheelchair users are aged 65 and older. Apart from aging, other factors that lead to increased use of wheelchairs are genetic disorders, accidents and other factors [1]. This is apparent from a survey showing that the average annual revenue growth rate in the wheelchair industry sector was 2.5% from 2009 to 2014.

Initially, traditional hand-operated wheelchairs had only two wheels attached to the chair which had to be manually operated by the user or another person. Many researchers have investigated and developed the technology for wheelchairs and there is still room for improvement. Some patients cannot to operate the wheelchair with their hands since they are unable to move most parts of their body, thus leading to the rise of automated wheelchairs.

One of the earliest developments of intelligent wheelchairs was the system where they placed a chair on a robot that provided mobility to the wheelchair [2]. The user could control it using joysticks installed on the armrests and linked to the robot. In addition, Nipanikar *et al.*, [3] used a voice

* *Corresponding author*
*E-mail address: asep.sholahuddin@unpad.ac.id*

https://doi.org/10.37934/ard.139.1.144157

recognition system to operate an automatic wheelchair and infrared sensor and ultrasonic systems to avoid obstacles. Although the use of voice recognition and voice commands to handle a wheelchair may sound innovative, it was inefficient and unreliable due to outdoors noise that can manipulate and even change the user's voice commands. In addition, using accelerometer sensors and joysticks provided for people with disabilities who could use their hands. In 2014, Pande *et al.,* [4] created a wheelchair operated by hand gesture recognition. It used an accelerometer sensor resulting in a wheelchair that could be controlled and could move freely in all directions. Sajeevan *et al.,* [5] designed a wheelchair that can be controlled by head movements. They made a using head orientation detection unit (Ultrasonic Sensors). It was able to detect the head's position through ultrasonic sensors and control the motor attached to both wheels. This was beneficial for patients who could only move their head. Balsara *et al.,* [6] designed a wheelchair controlled by an Android device such as a mobile phone. The wheelchair was controlled remotely through Bluetooth feature on the mobile device [7] and was operated by hand gestures using a gyroscope sensor to help tech-phobic people who were uncomfortable with phone functions, to reduce physical work and to assist in having an advanced and progressive life [8]. The wheelchair used battery power and Arduino microcontroller. The device consisted of two parts, consists of transmitter end and receiver using MPU-6050 gyro accelerometer sensor [9].

People who are limited to only being able to move their head need a tool in the form of a wheelchair that can be moved using the head. Controlling robot arms using a camera or Kinect-based head had been attempted but it was still complicated because it required using a camera and computer [10]. Then a wheelchair controller using brain waves was also developed but did not go well because there was a problem that if the mind was not focused then the robot arm would not move as desired [11]. The next development required using a Gyro sensor to detect head tilt [12]. Wheelchairs using the head to move have been developed using a Gyro sensor but there is still a problem that a lot of movement might cause the head to tire. Therefore, it is necessary to develop wheelchair movement using artificial intelligence that can automatically detect rooms in a building so that the wheelchair can move by itself to see the objects around it. For example, if someone wanted go to the living room from a bedroom, then the wheelchair would automatically move to look for the living room image object and if the wheelchair found the room, the wheelchair would turn to that room. In this research, YOLOv8 was used to produce a more accurate method in operating the wheelchair automatically.

## 2. Methodology

The basic method used in YOLOv8 is Convolution Neural Network (CNN). CNN is a method of Deep Learning [34-36]. The process carried out on CNN is by training numerous images and then testing it using the provided images. From the results of the CNN process, its accuracy was obtained. The CNN scheme is depicted in Figure 1.

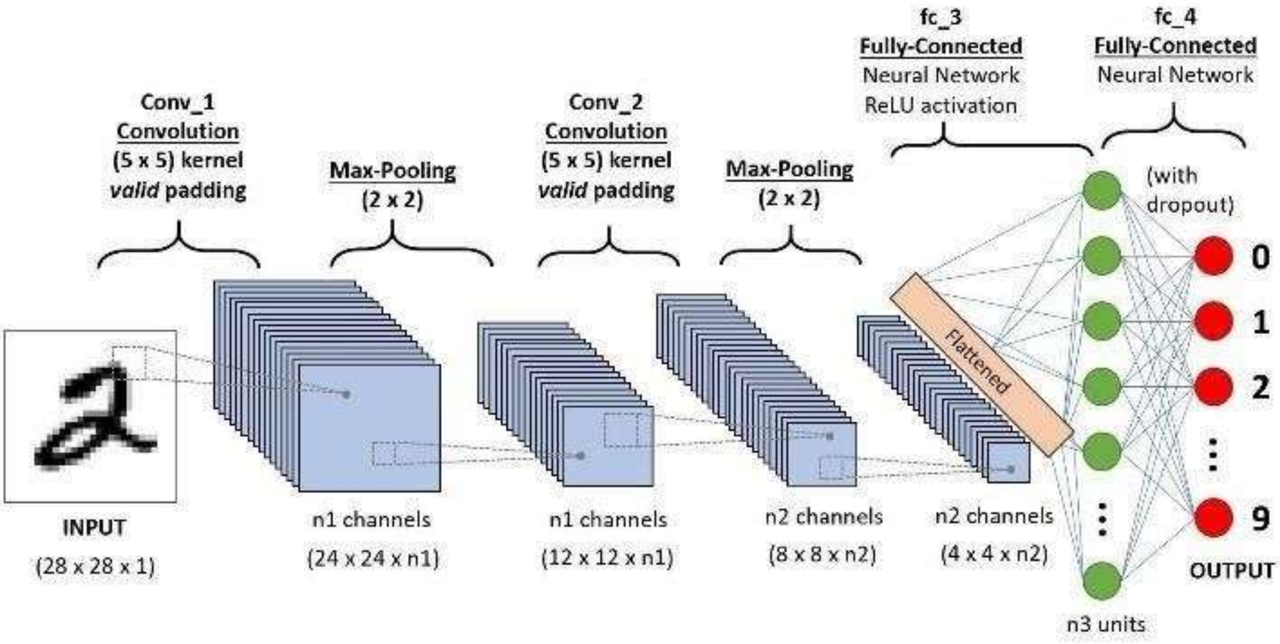



**Fig. 1.** CNN (Source: Sumit Saha -A comprehensive guide to convolutional neural networks -the ELI5 way)

### *2.1 Research Flow Chart*

In this research, the use of Artificial Intelligence (AI) is specifically for YOLOv8. In the YOLOv8 method, object recognition is better compared to the previous version. The systematics of this research in the form of a research flowchart is presented in Figure 2.

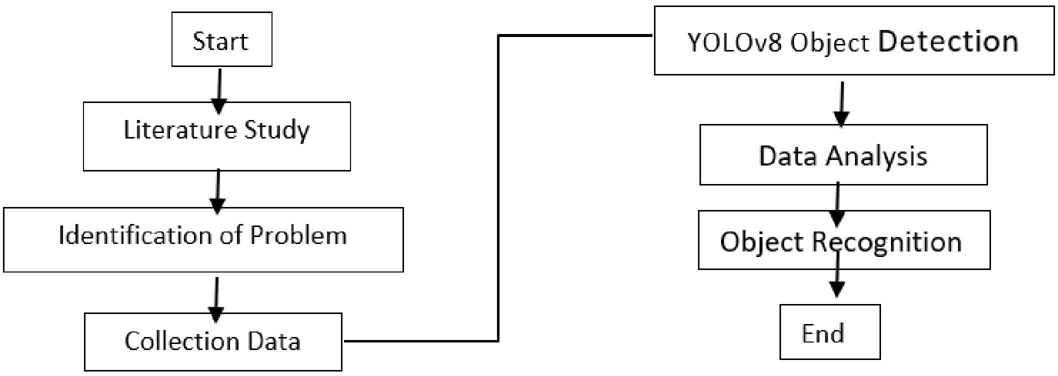



**Fig. 2.** Research stages

The research flowchart consists of six important stages which aim to achieve the research objectives comprehensively. The first stage was conducting an in-depth literature study and problem identification, establishing a foundation for comprehending existing knowledge and identifying spaces that would help disabled people using wheelchairs automatically navigate to those spaces.

After the literature study, the second stage was problem identification. The problem was to find the optimal value of automatic room search object recognition. The wheelchair would move to find the room it wanted to go to by recognizing the image object in front of the room. The research used 10 rooms using 10 types of symbols that were firstly trained using YOLOv8.

The third stage was collecting relevant data. This included collecting pictures that would be placed in front of the room so that the room could be recognized. There were 10 types of images that would be recognized and placed at the front of the room. The ten pictures were (1) bedroom, (2) toilet, (3) guest reception, (4) garage, (5) study room, (6) bathroom, (7) living room, (8) dining room, (9) prayer room and (10) garden. The room plan is presented in Figure 3.

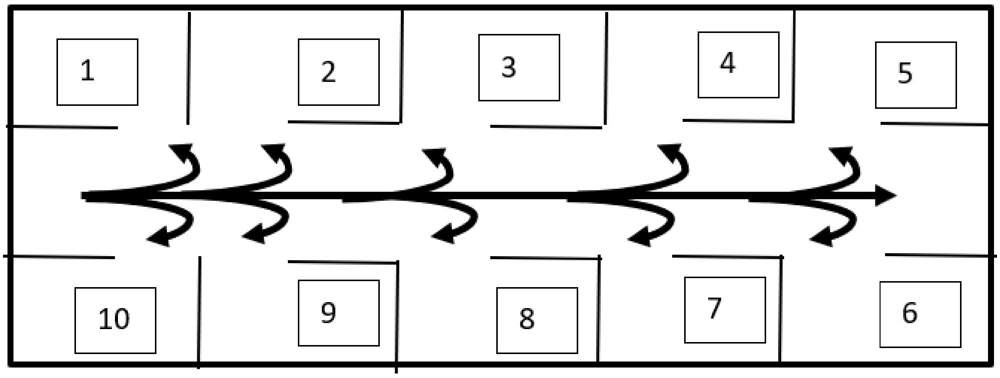



**Fig. 3.** Room schematics

In the fourth stage, the object detection for YOLOv8 was implemented. YOLOv8 used a Python program to recognize these symbols.

The fifth stage was the data analysis, namely the extent to how this method can be applied. These included Accuracy, Recall and F1 Score values resulting from the training carried out.

The sixth stage was implementing object recognition for each room. The results of room object recognition are be reported here.

In this research, a wheelchair was tested to explore rooms and then to search for the desired room by recognizing the images in front of the rooms. If the image being searched for was found, the wheelchair moved into the room. The images in front of the room were previously trained using YOLOv8.

*2.2 You Only Look Once Version 8 (YOLOv8)*

This research involved collecting data that included images and image recognition. Then, the YOLOv8 model was trained using an image dataset that included 10 types of images to recognize rooms and the training data included object annotations collected from field testing [13].

The results of measuring object recognition and system response time were collected during performance testing [14]. The information collected was assessed to ascertain how good this strategy helped automatic object recognition. Object detection movements can be analysed so that techniques can be updated and improved based on evaluation findings. The flowchart in Figure 4 illustrates that the steps in finding the object.

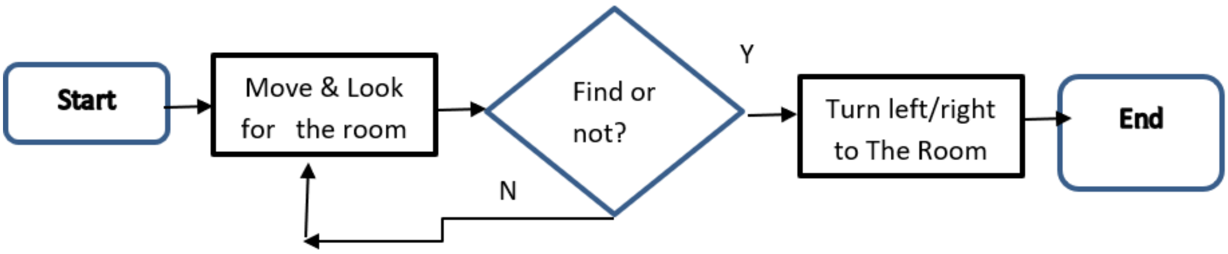



**Fig. 4.** Finding object

The sequence begins with camera initiation, followed by the object detection process by utilizing camera input. Next, there are steps to perform object detection and data training and refining the model's capabilities. The system then proceeds to distance detection, checking whether the object is successfully detected.

### *2.3 Object Recognition Formula*

The object recognition was based on the calculation results of the recognition percentage. Trials on the object recognition percentage were conducted a number of times to obtain a threshold value. The value was determined after the threshold value for the recognized objects was obtained.

## 3. Results

### *3.1 Results of Literature Studies*

The first stage was the was the literature study. YOLOv8, a deep learning model, was designed to detect objects. The model is capable of detecting multiple objects in one image frame at the same time. The YOLOv8 model is able to recognize objects in images or videos efficiently and accurately after rigorous training [15,31,32].

Table 1 presents the comparison between two real-time object detection models, YOLOv5 and YOLOv8, which involves a comprehensive evaluation specifically in supporting the mobility people with visual impairment. The two models are significantly different in terms of performance and careful consideration is required before used a model to detect objects. Albayrak [16] revealed that YOLOv8 had an impressive rate of accuracy of 94.2%, compared to YOLOv5 at 92.5%. Even though percentage difference seemed minor, YOLOv8 was able to detect objects more accurately. Regarding detection speed, YOLOv5 only needed 25 milliseconds per frame. However, YOLOv8 was a little slower at 28 milliseconds per frame. In a case that highspeed real-time detection is a crucial factor, YOLOv5 has an advantage [17].

GPU consumption was also included in the considerations. YOLOv5 was more efficient at 70% GPU resource usage, whereas YOLOv8 used 75%. In operations with limited GPU availability, the usage of GPU resources is an important consideration. Another key factor to consider was the size of the model and YOLOv5 had an advantage for being smaller in size which means efficiency in using storage space. On the contrary, YOLOv8, that had a larger model size, might require more space to store [18]. Yet another aspect to reflect was the training data set. YOLOv5 was claimed to have a firm dataset to train models for object detection, which implied YOLOv5's superior performance. However, YOLOv8 has a feature to continuously update its dataset, which emphasizes the commitment to do continuous improvement on the training dataset quality [19].

The decision to choose YOLOv5 or YOLOv8 was based on the priority of specific object detection applications. The main considerations for the selection were detection speed, accuracy, resource efficiency, model size and dataset update availability [16].

**Table 1**
Real-time detection comparison table

| Metric | YOLOv5 | YOLOv8 |
|---|---|---|
| Accuracy | 92,5 % | 94,2% |
| Detection Time | 25 ms/frame | 28 ms/frame |
| GPU consumption | 70% Usage | 75% |
| Model Size | Smaller | Bigger |
| Dataset Availability | Simply Perfect | Improved to the latest version |

YOLOv5, as object detection method, which stands for “You Only Look Once,” separates images into a grid structure. Each grid cell is in charge of detecting items within itself [20,33]. The intricacy of sentences and their contextual nuances significantly influences the accuracy of translations [21,34]. This current research prefers YOLOv8 due to higher accuracy compared to YOLOv5.

### *3.2 Problems to be Resolved*

The room used in this research was Building-D of Universitas Padjadjaran with a floor plan as shown in Figure 3. There were 10 rooms with 10 symbols that would be recognized. The wheelchair could move to find the room to go to by entering the named room. The movement of the wheelchair is presented in Figure 3, while its specifications are described in Table 2 and Figure 5.

**Table 2**
Specification of the wheelchair

| No. | Name | Specification | Unit |
|---|---|---|---|
| 1 | Wheelchair | • -Arduino Microcontroller | 1 |
| | | • -DC Motor | 2 |
| | | • -Driver Motor | 2 |
| | | • -Wheel motor | 2 |
| | | • -Battery | 1 |
| 2 | Computer | • -NUC | 1 |
| | | • -Webcam Logi 1080 | 2 |
| | | • -Monitor 7 “ | 1 |

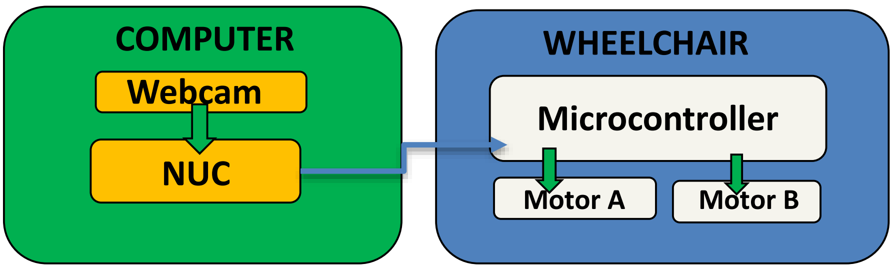


**Fig. 5.** Schematics of the wheelchair system

### *3.3 Data Collection and Annotation*

There were 10 types of data collected namely (1) bedroom (kamar tidur/bed), (2) toilet (closet/toilet), (3) guest reception/ruang tamu (lampu/lamp), (4) garage (mobil/car), (5) study room (buku/book), (6) bathroom (shower), (7) living room (sofa), (8) dining room (ruang makan/dining), (9) prayer room (masjid/mosque) and (10) garden (pohon/tree). The pictures were taken from the following sources and the symbol images are given in Figure 6 [22-31].

i. bedroom: https://www.freepik.com/icon/bedroom_2642268
ii. toilet: https://www.vectorstock.com/royalty-free-vector/toilet-color-design-concept-vector-21892785
iii. lamp: https://pngtree.com/freepng/white-background-with-black-icon-of-a-hanging-chandelier-lamp-vector_12888549.html
iv. car: https://www.pngdownload.id/png-jb4wlz
v. book: https://id.pngtree.com/freepng/book-icon_159611.html?share=3
vi. shower: https://www.svgrepo.com/svg/48096/shower
vii. sofa: https://www.flaticon.com/free-icon/sofa_2887417
viii. Dining: https://universe.roboflow.com/bigpp/fork_spoon_icon_detection
ix. mosque: https://id.pngtree.com/freepng/mosque-logo-vector-peace-creative_8175736.html
x. tree: https://www.shutterstock.com/image-vector/simple-tree-decor-silhouette-vector-260nw-1710053998.jpg

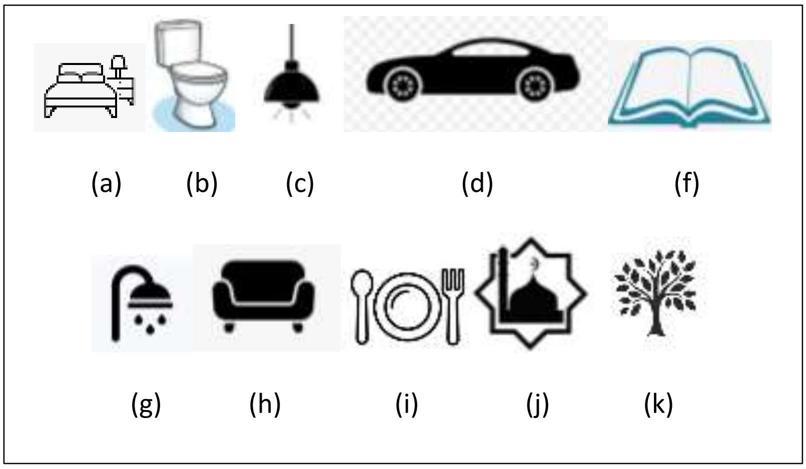



**Fig. 6.** Symbol for image recognition

The 10 images in Figure 6 were trained to the model in several steps. First, annotations were made by giving a box mark and labelling the name of the image. For example, the image of a toilet was made into a box and given the name “closet” (toilet), so that later if a wheelchair needs to go to the toilet, it will look for the image of the toilet in front of the room. Before training, the symbols were annotated to make recognition easier. The example of annotation is visualized in Figure 7.

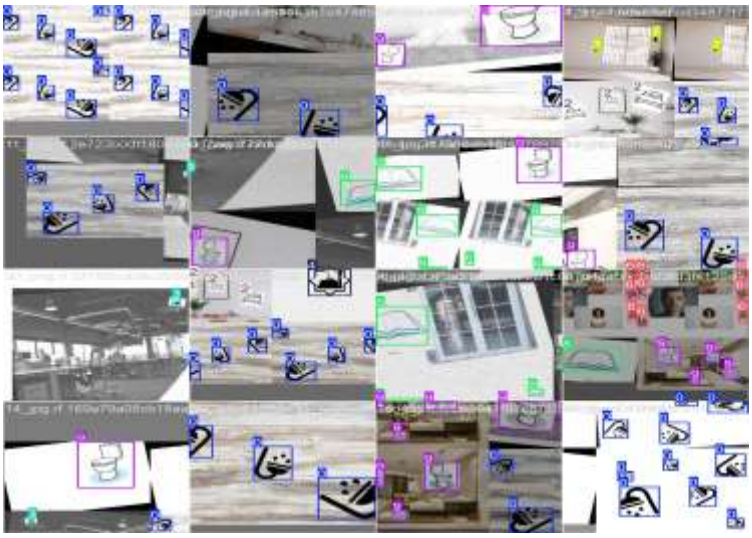


**Fig. 7.** The results of annotation

### *3.4 Implementation of YOLOv8*

The programming language used was Python while the library used YOLOv8 to detect objects. The 10 types symbols were made into a dataset of 2,494 images of different sizes and different positions. The initial image size was 760 x 1080 pixels. The dataset was trained and then tested using this program. Figure 8 shows the program's architecture.

| | from | n | params | module | arguments |
|---|---|---|---|---|---|
| 0 | -1 | 1 | 464 | ultralytics.nn.modules.conv.Conv | [3, 16, 3, 2] |
| 1 | -1 | 1 | 4672 | ultralytics.nn.modules.conv.Conv | [16, 32, 3, 2] |
| 2 | -1 | 1 | 7360 | ultralytics.nn.modules.block.C2f | [32, 32, 1, True] |
| 3 | -1 | 1 | 18560 | ultralytics.nn.modules.conv.Conv | [32, 64, 3, 2] |
| 4 | -1 | 2 | 49664 | ultralytics.nn.modules.block.C2f | [64, 64, 2, True] |
| 5 | -1 | 1 | 73984 | ultralytics.nn.modules.conv.Conv | [64, 128, 3, 2] |
| 6 | -1 | 2 | 197632 | ultralytics.nn.modules.block.C2f | [128, 128, 2, True] |
| 7 | -1 | 1 | 295424 | ultralytics.nn.modules.conv.Conv | [128, 256, 3, 2] |
| 8 | -1 | 1 | 460288 | ultralytics.nn.modules.block.C2f | [256, 256, 1, True] |
| 9 | -1 | 1 | 164608 | ultralytics.nn.modules.block.SPPF | [256, 256, 5] |
| 10 | -1 | 1 | 0 | torch.nn.modules.upsampling.Upsample | [None, 2, 'nearest'] |
| 11 | [-1, 6] | 1 | 0 | ultralytics.nn.modules.conv.Concat | [1] |
| 12 | -1 | 1 | 148224 | ultralytics.nn.modules.block.C2f | [384, 128, 1] |
| 13 | -1 | 1 | 0 | torch.nn.modules.upsampling.Upsample | [None, 2, 'nearest'] |
| 14 | [-1, 4] | 1 | 0 | ultralytics.nn.modules.conv.Concat | [1] |
| 15 | -1 | 1 | 37248 | ultralytics.nn.modules.block.C2f | [192, 64, 1] |
| 16 | -1 | 1 | 36992 | ultralytics.nn.modules.conv.Conv | [64, 64, 3, 2] |
| 17 | [-1, 12] | 1 | 0 | ultralytics.nn.modules.conv.Concat | [1] |
| 18 | -1 | 1 | 123648 | ultralytics.nn.modules.block.C2f | [192, 128, 1] |
| 19 | -1 | 1 | 147712 | ultralytics.nn.modules.conv.Conv | [128, 128, 3, 2] |
| 20 | [-1, 9] | 1 | 0 | ultralytics.nn.modules.conv.Concat | [1] |
| 21 | -1 | 1 | 493056 | ultralytics.nn.modules.block.C2f | [384, 256, 1] |
| 22 | [15, 18, 21] | 1 | 751507 | ultralytics.nn.modules.head.Detect | [1, [64, 128, 256]] |

Model summary: 225 layers, 3011043 parameters, 3011027 gradients, 8.2 GFLOPs

**Fig. 8.** The architecture of YOLOv8

### *3.5 Calculation Results for using YOLOv8*

The training and validation data sets of this program are presented in Figures 9 and 10. The total for training data and validation data combined was 2,494 (2,218 dan 276 respectively).


**Fig. 9.** Training set distribution

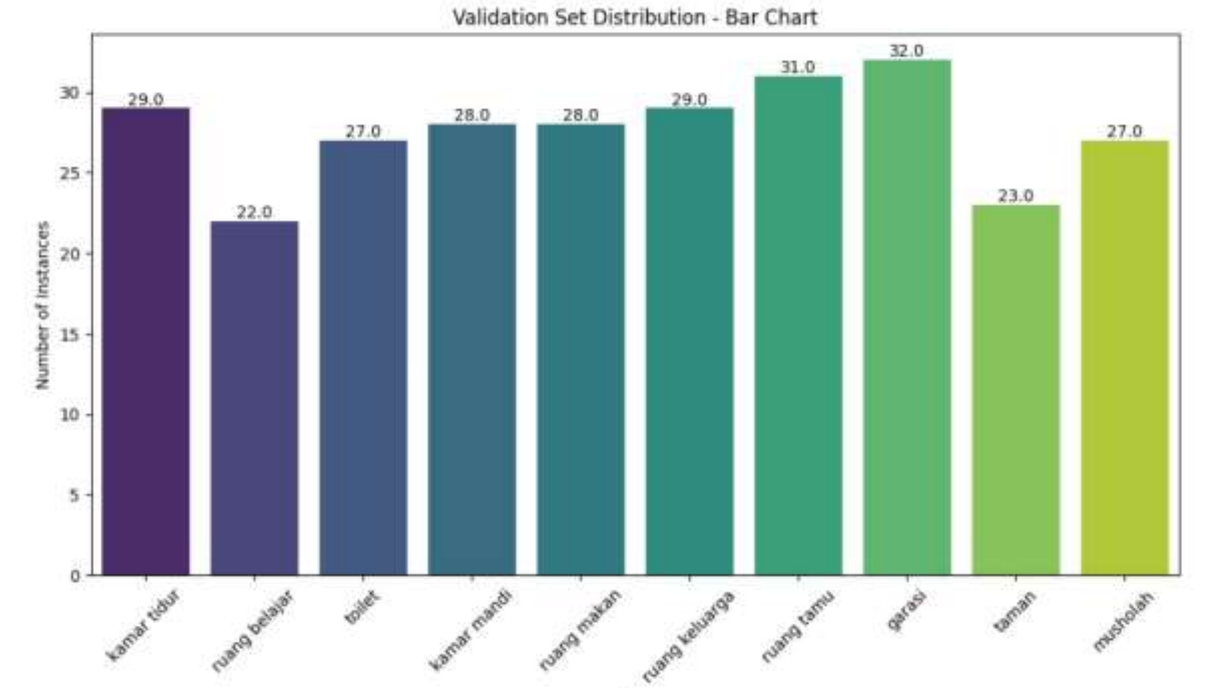
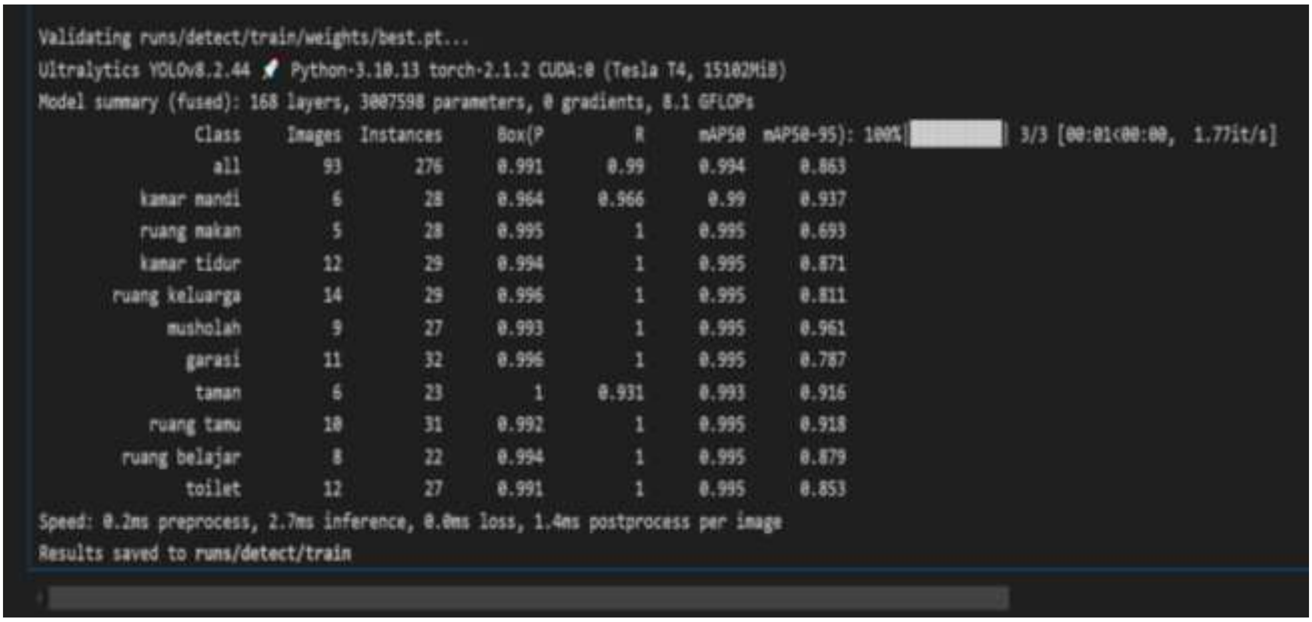



**Fig. 10.** Validation set distribution

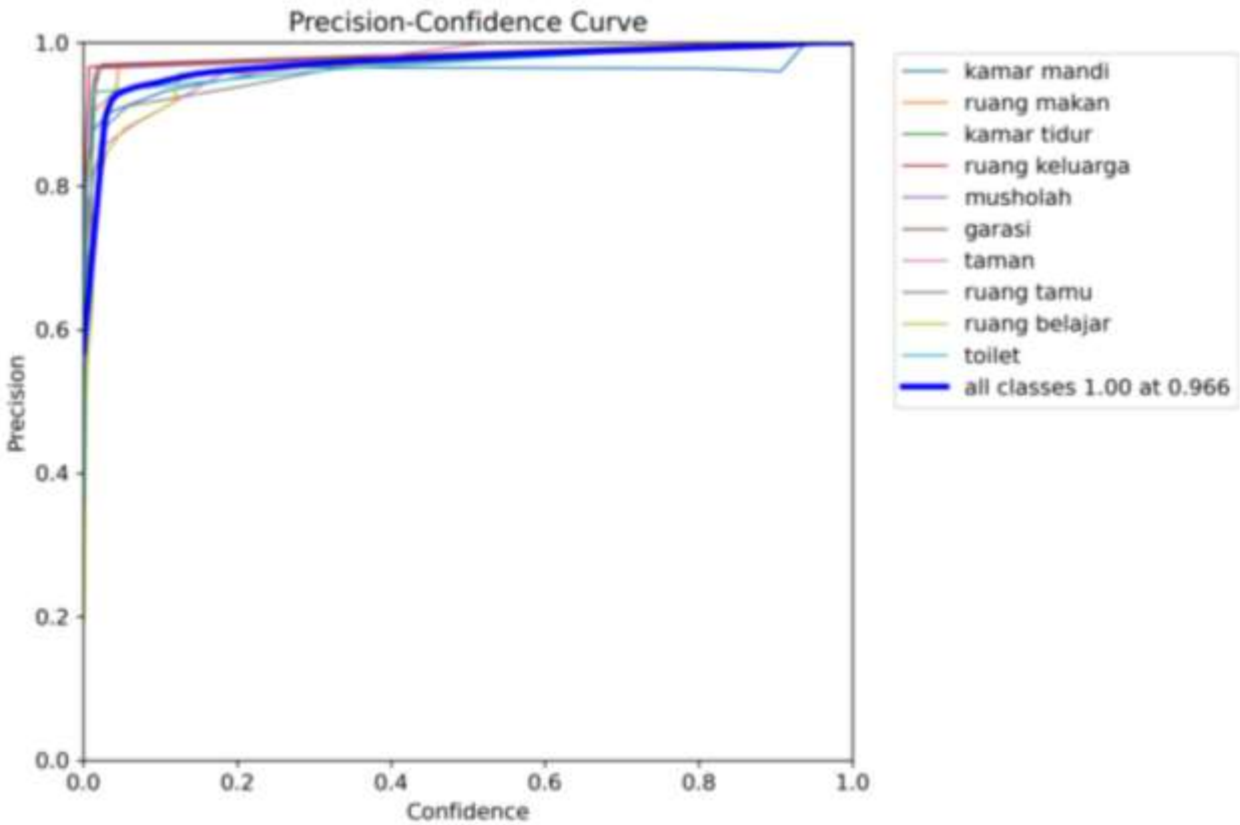
The summary of the phyton program shows that the average percentage of the recognition is more than 99% (mAP50) as shown Figure 11.



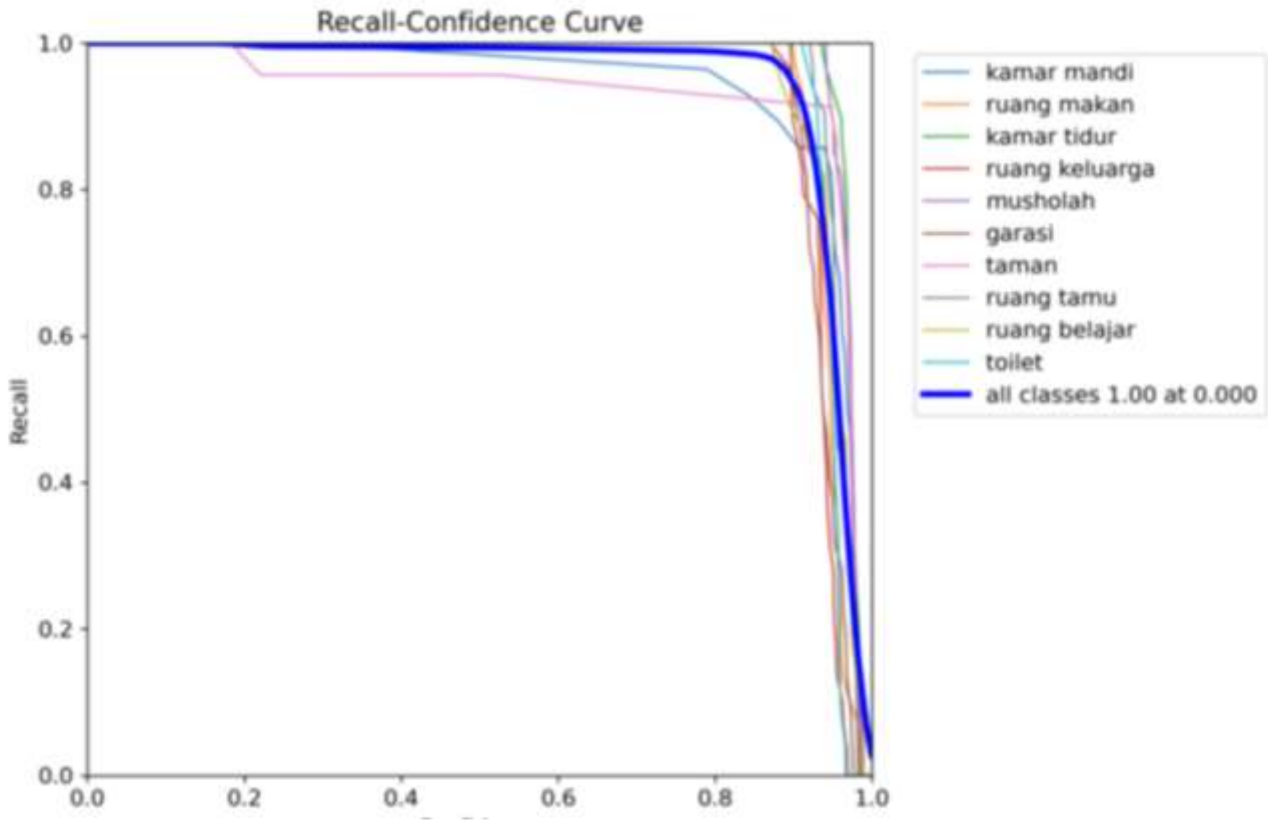
```
Validating runs/detect/train/weights/best.pt...
Ultralytics YOLOv8.2.44 🚀 Python-3.10.13 torch-2.1.2 CUDA:0 (Tesla T4, 15102MiB)
Model summary (fused): 168 layers, 3007598 parameters, 0 gradients, 8.1 GFLOPs
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100%|██████████| 3/3 [00:01<00:00,  1.77it/s]
                   all         93        276      0.991       0.99      0.994      0.863
           kamar mandi          6         28      0.964      0.966       0.99      0.937
           ruang makan          5         28      0.995          1      0.995      0.693
           kamar tidur         12         29      0.994          1      0.995      0.871
        ruang keluarga         14         29      0.996          1      0.995      0.811
              musholah          9         27      0.993          1      0.995      0.961
                garasi         11         32      0.996          1      0.995      0.787
                 taman          6         23          1      0.931      0.993      0.916
            ruang tamu         10         31      0.992          1      0.995      0.918
          ruang belajar          8         22      0.994          1      0.995      0.879
                toilet         12         27      0.991          1      0.995      0.853
Speed: 0.2ms preprocess, 2.7ms inference, 0.0ms loss, 1.4ms postprocess per image
Results saved to runs/detect/train
```

**Fig. 11.** Result of the recognition for 10 symbols

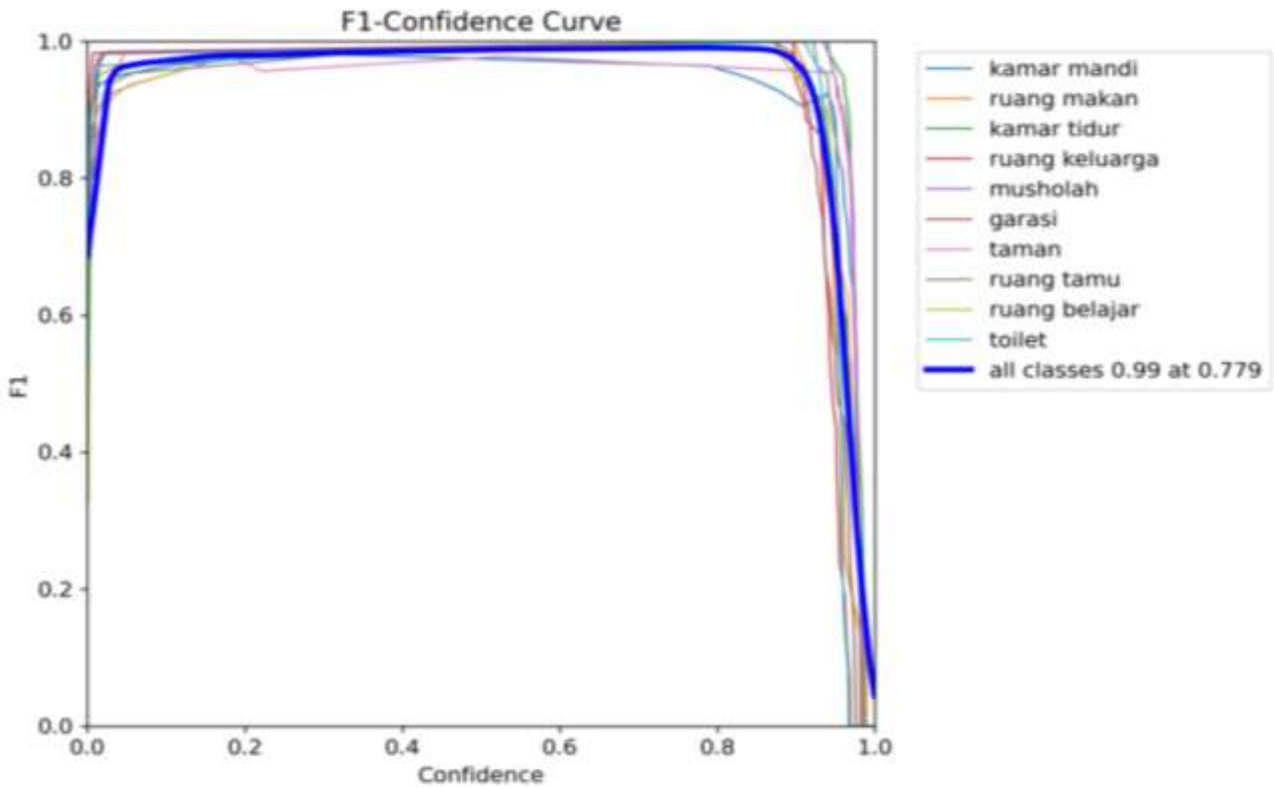
Figure 12 illustrates the results of accuracy, recall and F1 Score from the training process. The result of F1 is about 99%, which means 99% the program can recognize the images and that the training was successful.




(a)




(b)




(c)

**Fig. 12.** Results of (a) Precision (b) Recall (c) F1

The result of prediction is presented in Figure 13.

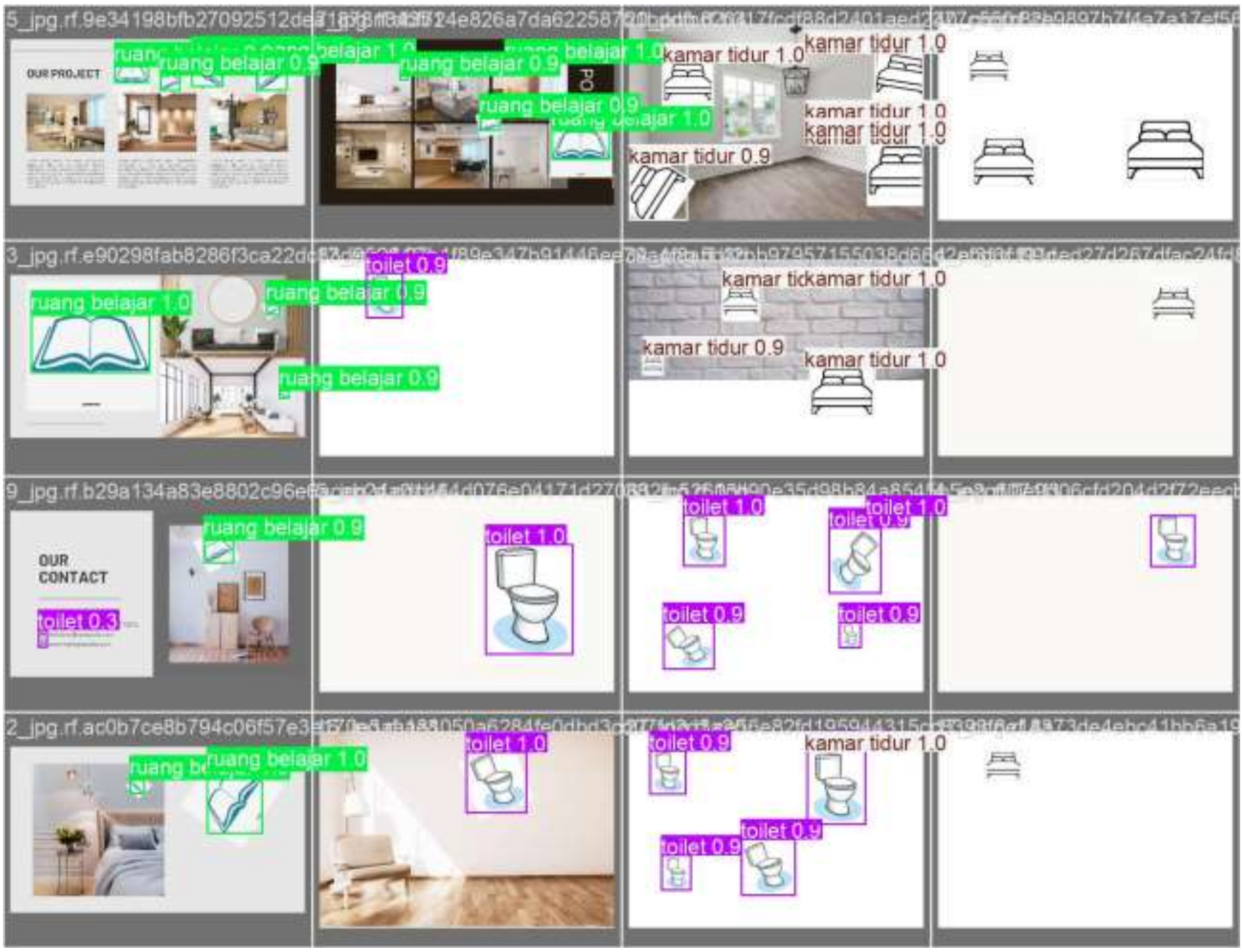



**Fig. 13.** The result of symbol prediction after training

### *3.6 Results of Room Recognition*

There are 10 rooms that were tested. Table 3 lists the results of object recognition to identify the room. The results shows that the recognition was successful.

**Table 3**
Result of recognition 10 rooms

| No. | Type of Room | Test | Result of Recognition | Status |
|---|---|---|---|---|
| 1. | Bedroom (kamar tidur/bed) | kamar tidur | kamar tidur | True |
| 2. | Toilet (closet/toilet) | closet | closet | True |
| 3. | Living room (lampu/lamp) | lampu | lampu | True |
| 4. | Garage (mobil/car) | mobil | mobil | True |
| 5. | Study room (buku/book) | buku | buku | True |
| 6. | Bathroom (shower) | shower | shower | True |
| 7. | Family room (sofa) | sofa | sofa | True |
| 8. | Dining room (ruang makan/dining) | ruang makan | ruang makan | True |
| 9. | Prayer room (masjid/mosque) | masjid | masjid | True |
| 10. | Garden (pohon/tree) | pohon | pohon | True |

## 4. Conclusion

The program language used was Python programming with the CNN method, and YOLOv8 model. The object recognition in the room was accomplished. During training with a total more than 2,400 images of data from 10 types of image symbols used, an accuracy/precision value of 100% was obtained with a confidence rate of 0.996, a Recall value of 100% with confidence rate of 0.0 and an F1 Score value of 99% with a confidence rate of 0.779. These results showed that the object recognition of the was a success. Direct recognition of objects in the field resulted in an average detection rate of 100%, thus the wheelchair can effectively find the correct room.


## Acknowledgement
This research was funded by a grant from RPLK (Accelerated Research for Associate Professors) of Universitas Padjadjaran. The Contract Number is 1706/UN6.31/PT.00/2024.


## References

[1] World Health Organization. "Guidelines on the provision of manual wheelchairs in less resourced settings." In *Guidelines on the provision of manual wheelchairs in less resourced settings*. 2008.

[2] Gomi, Takashi and Ann Griffith. "Developing intelligent wheelchairs for the handicapped." *Assistive technology and artificial intelligence: Applications in robotics, user interfaces and natural language processing* (2006): 150-178. https://doi.org/10.1007/BFb0055977

[3] Nipanikar, R. S., Vinay Gaikwad, Chetan Choudhari, Ram Gosavi and Vishal Harne. "Automatic wheelchair for physically disabled persons." *Int. J. Adv. Res. Electron. Commun. Eng* 2, no. 4 (2013): 466-478.

[4] Pande, Vishal V., Nikita S. Ubale, Darshana P. Masurkar, Nikita R. Ingole and Pragati P. Mane. "Hand gesture based wheelchair movement control for disabled person using MEMS." *International Journal of Engineering Research and Applications* 4, no. 4 (2014): 152-158.

[5] Sajeevan, Rishi Prasad, Fasil Yousaf, Basil James and Jisha Kuruvilla. "Head Orientation Controlled Wheelchair." *International Research Journal of Engineering and Technology* 4, no. 4 (2017): 1474–78. https://www.irjet.net/archives/V4/i4/IRJET-V4I4303.pdf

[6] Balsara, M. D., V. S. Takate and Bachkar Siddhant. "M. D. Balsara." *International Journal of Advances in Scientific Research and Engineering* 3, no. 4 (2017): 47–50. https://doi.org/10.31695/IJASRE

[7] Rai, Nelson, Deepak Rasaily, Tashi Rapden Wangchuk, Manoj Gurung and Rit Kr Khawas. "Bluetooth remote controlled car using Arduino." *Int. J. Eng. Trends Technol* 33 (2016): 381-384. https://doi.org/10.14445/22315381/IJETT-V33P274

[8] Al-Neami, Auns QH and Saba M. Ahmed. "Controlled wheelchair system based on gyroscope sensor for disabled patients." *Biosciences Biotechnology Research Asia* 15, no. 4 (2018): 921. https://doi.org/10.13005/bbra/2703

[9] Farhan Razy, Md, Sabah Shahnoor Anis, Md Touhiduzzaman Touhid and Jia Uddin. "An automated wheelchair for physically challenged people using hand gesture and mobile app." In *Intelligent Human Computer Interaction: 12th International Conference, IHCI 2020, Daegu, South Korea, November 24–26, 2020, Proceedings, Part II 12*, pp. 135-144. Springer International Publishing, 2021. https://doi.org/10.1007/978-3-030-68452-5_13

[10] Sholahuddin, Asep, Setiawan Hadi and Muhamad Indra Kusmana. "Pergerakan Kursi Roda Menggunakan Kepala Melalui Kinect (Wheelchair Movement Using the Head through Kinect)." *In Prosiding Seminar Nasional Matematika Tahun 2015 (Proceedings of the 2015 National Mathematics Seminar)*, edited by Endang Rusyaman, Juli Rejito, Rudi Rosadi and Jordy Saragih, 183–89. Bandung-Sumedang: Universitas Padjadajaran, (2016).

[11] Sholahuddin, Asep, Setiawan Hadi and Lany Rahmawati. "Application Brain Wave for Wheel Robotic Movement Using Mindflex." *Integrated Sci-Tech: The Interdisciplinary Research Approach*, Volume 1, edited by Ardian Ulvan and Irza Sukmana, 222–27. Bandar Lampung: UPT. Perpustakaan Universitas Lampung, (2015). http://digilib.unila.ac.id/21632/40/BC_final_IC-STAR2015_MSBE_Chapter 30.pdf

[12] Kumar, Debasis and Naveed Muhammad. "Object detection in adverse weather for autonomous driving through data merging and YOLOv8." *Sensors* 23, no. 20 (2023): 8471. https://doi.org/10.3390/s23208471

[13] Fadlilah, Umi and Bana Handaga. "The development of android for Indonesian sign language using tensorflow lite and CNN: an initial study." In *Journal of Physics: Conference Series*, vol. 1858, no. 1, p. 012085. IOP Publishing, 2021. https://doi.org/10.1088/1742-6596/1858/1/012085

[14] Liu, Kehong. "Stbi-yolo: A real-time object detection method for lung nodule recognition." *IEEE Access* 10 (2022): 75385-75394. https://doi.org/10.1109/ACCESS.2022.3192034

[15] Arifando, Rio, Shinji Eto and Chikamune Wada. "Improved YOLOv5-based lightweight object detection algorithm for people with visual impairment to detect buses." *Applied Sciences* 13, no. 9 (2023): 5802. https://doi.org/10.3390/app13095802
[16] Albayrak, Nahuel E. "Object recognition using TensorFlow." In *2020 IEEE integrated STEM education conference (ISEC)*, pp. 1-1. IEEE, 2020. https://doi.org/10.1109/ISEC49744.2020.9397835
[17] Lin, Cheng-Jian, Shiou-Yun Jeng and Hong-Wei Lioa. "A Real-Time Vehicle Counting, Speed Estimation and Classification System Based on Virtual Detection Zone and YOLO." *Mathematical Problems in Engineering* 2021, no. 1 (2021): 1577614. https://doi.org/10.1155/2021/1577614
[18] Bacea, Dan-Sebastian and Florin Oniga. "Single stage architecture for improved accuracy real-time object detection on mobile devices." *Image and Vision Computing* 130 (2023): 104613. https://doi.org/10.1016/j.imavis.2022.104613
[19] Bochkovskiy, Alexey, Chien-Yao Wang and Hong-Yuan Mark Liao. "Yolov4: Optimal speed and accuracy of object detection." *arXiv preprint arXiv:2004.10934* (2020).
[20] Vieira, Lucas Nunes, Minako O'Hagan and Carol O'Sullivan. "Understanding the societal impacts of machine translation: a critical review of the literature on medical and legal use cases." *Information, Communication & Society* 24, no. 11 (2021): 1515-1532. https://doi.org/10.1080/1369118X.2020.1776370
[21] Max.icons. "Bedroom Icon." *Freepik*. https://www.freepik.com/icon/bedroom_2642268
[22] Nickylarson974. "Toilet Color Design Concept Vector Image." *Vector Stock*. https://www.vectorstock.com/royalty-free-vector/toilet-color-design-concept-vector-21892785
[23] Yayimage. "White Background with Black Icon of a Hanging Chandelier Lamp Vector." *Pngtree*. https://pngtree.com/freepng/white-background-with-black-icon-of-a-hanging-chandelier-lamp-vector_12888549.html
[24] Emeodi. "Mobil, Mobil Sport Png." *Pngdownload.id*. https://www.pngdownload.id/png-jb4wlz
[25] CCCC. "Ikon Buku." *Pngtree*. https://id.pngtree.com/freepng/book-icon_159611.html?share=3
[26] SVG Repo. "Shower Svg Vector." *SVG Repo*. https://www.svgrepo.com/svg/48096/shower
[27] Torskaya. "Sofa Free Icon." *Flaticon*. https://www.flaticon.com/free-icon/sofa_2887417
[28] Bigpp. "Fork_spoon_icon_detection Computer Vision Project." *Roboflow*. https://universe.roboflow.com/bigpp/fork_spoon_icon_detection
[29] Ingimage. "Logo Masjid Vektor Perdamaian Kreatif." *Pngtree*. https://id.pngtree.com/freepng/mosque-logo-vector-peace-creative_8175736.html
[30] 4LUCK. "Pohon Sederhana Dekorasi Siluet Vektor Gambar," https://www.shutterstock.com/id/image-vector/simple-tree-decor-silhouette-vector-image-1710053998
[31] Liu, Minggao, Ming Zhang, Xinlan Chen, Chunting Zheng and Haifeng Wang. "YOLOv8-LMG: an improved bearing defect detection algorithm based on YOLOv8." *Processes* 12, no. 5 (2024): 930. https://doi.org/10.3390/pr12050930
[32] Wang, Hai, Chenyu Liu, Yingfeng Cai, Long Chen and Yicheng Li. "YOLOv8-QSD: An improved small object detection algorithm for autonomous vehicles based on YOLOv8." *IEEE Transactions on Instrumentation and Measurement* (2024). https://doi.org/10.1109/TIM.2024.3379090
[33] Zhang, Yu, Zhongyin Guo, Jianqing Wu, Yuan Tian, Haotian Tang and Xinming Guo. "Real-time vehicle detection based on improved yolo v5." *Sustainability* 14, no. 19 (2022): 12274. https://doi.org/10.3390/su141912274
[34] Sary, Indri Purwita, Safrian Andromeda and Edmund Ucok Armin. "Performance comparison of YOLOv5 and YOLOv8 architectures in human detection using aerial images." *Ultima Computing: Jurnal Sistem Komputer* 15, no. 1 (2023): 8-13. https://doi.org/10.31937/sk.v15i1.3204
[35] Riad, Rachid, Olivier Teboul, David Grangier and Neil Zeghidour. "Learning strides in convolutional neural networks." *arXiv preprint arXiv:2202.01653* (2022). https://doi.org/10.31219/osf.io/4yz8f
[36] Krizhevsky, Alex, Ilya Sutskever and Geoffrey E. Hinton. "ImageNet classification with deep convolutional neural networks." *Communications of the ACM* 60, no. 6 (2017): 84-90. https://doi.org/10.1145/3065386