Enhancing Smart Wheelchair Navigation through Head Motion Detection: A YOLO-based Approach

Fitri Utaminingrum [1,*], Erlinda Butarbutar [1], Rahma Nur Fitriyani [1], Yvette Celia Aviani [1], and Atik Qomariyah [2]

[1] Department of Computer Science, Faculty of Computer Science, Brawijaya University, Malang, Indonesia
[2] Department of Nursing, Faculty of Health Sciences, Brawijaya University, Malang, Indonesia
Email: f3_ningrum@ub.ac.id (F.U.); erlinda@student.ub.ac.id (E.B.); rahmanur24@student.ub.ac.id (R.N.F.); yvettecelia@student.ub.ac.id (Y.C.A.); atikqomariyah_@student.ub.ac.id (A.Q.)
*Corresponding author

***Abstract*—In 2020, it was recorded that 5% of the total population in Indonesia were people with disabilities. Individuals with physical disabilities, especially those who cannot use both their hands and feet are facing problem in their mobility. Since most wheelchairs are controlled using hands, these individuals are unable to operate a wheelchair independently. This research aims to create smart wheelchairs that use object detection models, to enable the user to navigate their wheelchair. The smart wheelchair is equipped with a camera that will capture the user's head movement and will move based on it. Deep learning model algorithms are used to detect the head movement. In this research, three generations of the YOLO (You Only Look Once) model—YOLOv5, YOLOv6, and YOLOv7—are compared to determine the most suitable model for the system. It is found that YOLOv6N has the fastest inference time, that is 2.54 ms. All the models are also evaluated on several parameters: Precision, recall, mAP@.5, and mAP@.5:.95. There's no huge difference between each variation. All of the precision, recall and mAP@.5 of each variation are above 0.9. Yet, the difference can be seen for the mAP@.5:.95 where the highest score is 0.808 from YOLOv6L and the lowest is 0.703 from YOLOv5N.**

***Keywords*—deep learning, YOLO, head motion, detection, wheelchair, computer vision**

I. INTRODUCTION

In 2020, it was recorded that 5% of Indonesia's total population consisted of people with disabilities [1]. Disabilities are divided into several types, one of which is physical disability. Individuals with physical disabilities experience limitations or impairments in bodily movement functions. This condition can be caused by several factors such as congenital factors or accidents that occur before birth (prenatal phase), during birth (natal phase), and after birth (post-natal phase) [2].

People with physical disabilities often face problems and challenges in carrying out daily activities. The difficulty of mobility or movement means that people with physical disabilities need assistive devices to help them with mobility. People with physical disabilities, especially those with hands and feet that cannot function, need assistive devices such as wheelchairs for mobility.

Wheelchairs are divided into two, namely manual wheelchairs and electric wheelchairs. A manual wheelchair is a wheelchair that is operated by pushing the wheels on the right and left sides. Meanwhile, electric wheelchairs are wheelchairs that are generally controlled using a joystick. Both wheelchairs require hands that can function to control the wheelchair. This will cause a problem for people with physical disabilities who have problems with their hands.

A number of advances have been investigated to fill this gap. One of them is the use of Electroencephalogram (EEG) impulses to operate wheelchairs through brain activity. Although it has potential, this approach has several limitations, including the need for multiple electrodes, susceptibility to noise, and user discomfort during prolonged use. Moreover, EEG-based control often requires extensive calibration and user training, which can be time-consuming, making it less practical for real-world applications.

Similarly, alternative control methods, such as piezoelectric sensors, offer a different approach by detecting muscle movement or pressure variations. However, these methods can also pose challenges, such as discomfort from prolonged skin contact, potential inaccuracies due to unintentional muscle activation, and limitations in real-time response. Piezoelectric sensors may experience sensitivity degradation over time due to material fatigue or intensive use, necessitating periodic replacement. This weakness highlights the need for more intuitive and efficient control mechanisms that ensure reliable mobility assistance without sacrificing user comfort or daily functionality.

Manuscript received October 31, 2024; revised December 25, 2024; accepted February 14, 2025, published July 17, 2025.

doi: 10.18178/joig.13.4.325-335

Deep learning, a subset of machine learning, employs Artificial Neural Networks (ANNs) to model and solve complex problems. One advantage of deep learning is its ability to automatically learn patterns and make predictions based on data. This enhances performance in tasks such as image recognition, speech processing, and natural language understanding. Additionally, deep learning models can generalize well to new situations by learning abstract and hierarchical representations of data [3].

In several fields, deep learning has become a potent instrument for automated analysis and categorization. Additionally, studies on Convolutional Neural Networks (CNNs) for Image Detection and Recognition have shown how effective CNNs are in processing visual data, which makes them a fundamental method for a variety of image-based applications. Further demonstrating the versatility of YOLO-based models in practical recognition tasks, "Research on Car License Plate Recognition Based on Improved YOLOv5m and LPRNet" supports their promise for high-accuracy object detection and classification. These developments show how deep learning can be applied more broadly in vision-based systems, such as assistive technology for wheelchair navigation. YOLO (You Only Look Once) is a deep learning algorithm that uses Convolutional Neural Networks for object detection. Unlike other algorithms, the YOLO algorithm can reduce the computational power and time needed for training and inferring objects [4].

The proposed paper aims to assist the mobility of persons with disabilities while also contributing to the advancement of health technology in Indonesia by utilizing science and technology, particularly by implementing the latest methods in the field of Computer Vision.

II. Literature Review

Several innovations have been developed to address this issue. One of them is the innovation made by Landu Jiang *et al.* [5]. This innovation utilizes EEG (electroencephalogram) signals to record brain electrical activity captured from electrodes attached to the head, IMU sensors, and cameras to move wheelchairs [5]. It presents a BCI-based smart wheelchair control system that leverages EEG signals and motion sensing techniques to provide intuitive human-machine interaction for people with disabilities. To operate the wheelchair, users must focus on navigating. However, the number of electrodes attached to the user's head can cause discomfort. In addition, improper electrode placement may lead to inaccurate biomedical signal recordings. This impracticality can make it difficult for the user to control the wheelchair.

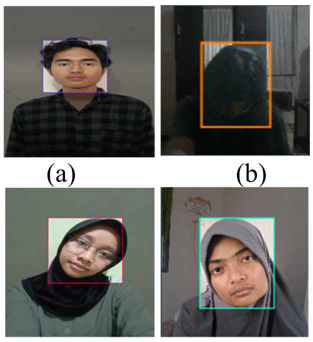
Another innovation is made by Charoenporn Bouyam and Yunyong Punsawad. This innovation utilizes piezoelectric sensors to obtain facial muscle signals to observe position when blinking the eyes and moving the tongue [6]. Piezoelectric sensors can convert physical quantities, i.e. acceleration, strain, force, or pressure into electrical signals without an external power supply. Thin layer sensors are tiny and sensitive, they are used in high-frequency applications. In applying the sensor, wheelchair users are required to control the movement of the tongue and blinking of the eyes periodically. However, it's considered ineffective because it limits the user's speaking ability. Eye blinking, which is a reflex movement, can confuse the navigation direction of a wheelchair. The placement of electrodes on the face will also interfere with user comfort.

III. Materials and Methods

This section discusses the implementation of the YOLO algorithm and CNN method to detect the direction of head movement.

A. Dataset

This research utilizes an image dataset consisting of 2224 images. The data was acquired using a standard camera with a fixed setup, ensuring consistency in image quality and angle. During the image collection process, the distance between the participant's face and the camera was maintained uniformly, and variations in head orientation, accessories, and data collection locations were considered to introduce diversity. The images are then divided into four classes. The four classes are front, down, right, and left. The sample image of the dataset can be seen in Fig. 1.



(a) (b) (c) (d)

Fig. 1. (a) front class (b) down class (c) right class (d) left class.

The dataset is further divided into two categories: training data and testing data. The training data is used to train the model, while the testing data evaluates its performance. The details of the dataset used can be seen in Table I. To support further research in this field, we have made the dataset publicly accessible via Roboflow at https://universe.roboflow.com/pkm-celia/deteksi-arah-gerak-kepala.

TABLE I. Persebaran Data

Jenis	Depan	Bawah	Kanan	Kiri
Latih	483	431	422	440
Validasi	81	118	149	99
Total	564	549	571	539

B. YOLOv5

YOLOv5 is the fifth generation of object detection model called "You Only Look Once" which is designed to provide real-time results with high speed and accuracy [7].

The model consists of four variances known as YOLOv5s, YOLOv5m, YOLOv5l, and YOLOv5x. There is no difference between the variances structure-wise. The difference is in the width and depth of the network [8]. The model works by taking the results from k-mean anchors and passing them through thousands of actual cost functions where the model will be trained [9].

YOLOv5N is the smallest model of YOLOv5. It has small parameters and relatively low hardware requirements [10]. The model has a good balance between size, speed and accuracy, which is good for mobile devices or embedded systems. Yet, the training process needs a longer time and higher hardware configuration.

YOLOv5S is a lightweight variation of YOLOv5, with 27M weight data. It is equal to 1/9 of YOLOv4 size, which allows this model to detect objects faster [11]. There are three parts of YOLOv5s, backbone, neck, and detection head [12]. Based on official information from the Ultralytics website, YOLOv5S is capable of processing images within 0.007 seconds. This is enough for real-time detection. Yet, YOLOv5S needs a large amount of training and capable hardware.

YOLOv5M is a YOLOv5 variation that is deeper and wider than YOLOv5S. It has a complex structure yet high detection accuracy. The network of YOLOv5M consists of four parts: input, backbone, neck, and prediction [13]. The complexity of the model made it hard to be trained and optimized.

YOLOv5L is a big detection model in all YOLOv5 variants. It has strong identification and extraction ability, high detection accuracy, and the ability to learn and quickly adapt [14]. This ability made YOLOv5l ideal for detecting big objects such as vehicles.

The complete comparison of four types of YOLOv5 models can be seen in Table II.

TABLE II. DIFFERENCE BETWEEN YOLOV5 VARIANCES [12]

Model	mAP (0.5)	mAP (0.5:0.95)	Speed v100 (ms)	Params (M)	FLOPs (G)
YOLOv5N	45.7	28.0	0.6	1.9	4.5
YOLOv5S	56.8	37.4	0.9	7.2	15.6
YOLOv6M	64.1	45.4	1.7	21.2	49.0
YOLOv6L	67.3	49.0	2.7	46.5	109.1

C. YOLOv6

YOLOv6 algorithm built on YOLO architecture with some improvements compared to previous YOLO versions. YOLOv6 has better performance in detecting small objects [15]. It integrates modern quantization techniques such as QAT (Quantization-Aware Training) and PTQ (Post-Training Quantization). Which is done to improve inference speed without lowering the network performance [16].

YOLOv6 also has four modifications such as renewing the network to the size compatible with the industry scenario with the best trade-off between accuracy and speed, adding a self-distillation strategy for classification and regression task, increasing performance by verifying detection technique to decide label, lost function, and data augmentation. Lastly, it reforms the quantization scheme to detect objects with the help of a RepOptimizer and channel-wise-distillation [17].

YOLOv6 provides four pre-trained models with different variations and scales. The four models are YOLOv6-N (nano), YOLOv6-S (small), YOLOv6-M (medium), and YOLOv6-L (large). The difference between the four models can be seen in Table III.

TABLE III. PERFORMANCE DIFFERENCES BETWEEN YOLOV6 VARIANCES [18]

Model	Size	mAP (0.5:0.95)	SpeedT4 trt fp16 b1 (fps)	SpeedT4 trt fp16 b32 (fps)	Parameters (M)	FLOPs (G)
YOLOv6N	640	37.5	779	1187	4.7	11.4
YOLOv6S	640	45.0	339	484	18.5	45.3
YOLOv6M	640	50.0	175	226	34.9	85.8
YOLOv6L	640	52.8	98	116	59.6	150.7

D. YOLOv7

YOLOv7 is also an object detection model. This model is better in speed and accuracy compared to the other object detection models such as YOLOv5, YOLOR, YOLOX, and others [19]. YOLOv7 focuses on the optimization of the training process, including modules and the concept “trainable bag-of-freebie”. It’s an optimization method that is designed to improve accuracy without increasing the inference cost [20]. There are six types of YOLOv7. Those types are YOLOv7, YOLOv7-X, YOLOv7-W6, YOLOv7-E6, YOLOv7-D6, dan YOLOv7-E6E.

YOLOv7 is a base model that is efficient on standard GPU training. Neck scale modeling and compound scaling is done to improve the model’s depth and width [21]. This improvement creates YOLOv7’s variants, YOLOv7-X.

YOLOv7-X is a YOLOv7 variant that is stronger and more advanced. This variation has more layers and parameters that allow it to process more information and more precise and accurate detection results [22]. This made YOLOv7-X need higher computation resources, which limits the usage to devices with low computational resources such as IoT and GPU Edge.

YOLOv7-W6 is a variation that is lightweight and efficient in object detection. This YOLOv7 variation has a smaller network architecture so it has smaller parameters and model size. YOLOv7-W6 has been optimized for cloud GPU Computing [21].

YOLOv7-E6, YOLOv7-D6, and YOLOv7-E6E is the optimization of YOLOv7-W6. This is done to get high-end GPU clouds [23]. YOLOv7-E6 produced by improvement using E-ELAN (Extended Efficient Layer Aggregation Network) to detect small objects. YOLOv7-D6 improves performance in some image sizes by compound multiscale method. YOLOv7-E6E is the result of depth and width improvements of YOLOv7-E6. This model is the biggest and most accurate model, yet has slow speed [24].

YOLOv7 performance comparison can be seen in Table IV.

TABLE IV. PEROFORMANCE DIFFERENCES BETWEEN YOLOV7 VARIANCES [25]

Jenis	Test Size	AP^{test} (%)	$AP_{50}{}^{test}$ (%)	$AP_{75}{}^{test}$ (%)	Batch 1 (fps)	Batch 32 avarage time (ms)
YOLOv7	640	51.4	69.7	55.9	161	2.8
YOLOv7-X	640	53.1	71.2	57.8	114	4.3
YOLOv7-W6	1280	54.9	72.6	60.1	84	7.6
YOLOv7-E6	1280	56.0	73.5	61.2	56	12.3
YOLOv7-D6	1280	56.6	74.0	61.8	44	15.0
YOLO-E6E	1280	56.8	74.4	62.1	36	18.7

E. Convolutional Neural Network (CNN)

CNN (Convolutional Neural Network) is a deep learning algorithm designed for image processing, utilizing filters (kernels) to extract relevant features [26]. The CNN algorithm consists of several layers which are divided into five types, namely input layer, hidden layer, output layer, convolutional layer, and pooling layer. The structure of the CNN algorithm is composed of three-dimensional neurons, namely width, height, and depth [27]. In CNN, height and width represent the size of layers while depth represents the number of layers.

F. Dataset Preprocessing

Data preprocessing is a process done to the dataset before it goes through the training phase. In this research, the preprocessing steps applied to the dataset included resizing and auto-contrast adjustments to standardize image dimensions and enhance contrast for better model performance..

G. Model Training

The YOLO model was trained on a dataset with four classes (front, down, right, left) using a batch size of 32, a 416×416 pixel input resolution, and 100 epochs. Annotations were stored in a YAML file, and training was performed on a single GPU. Performance was evaluated using Precision, Recall, mAP@.5, and mAP@.5:.95 on an independent dataset. The best checkpoint was selected for inference with a confidence threshold of 0.50.

H. Procedure

The system started by collecting images that will be used to train the model. Then, the images were annotated to make them a complete dataset. The images in the dataset then go through preprocessing. Here the preprocessing processes used were resize and auto contrast. The preprocessed data is then used to train the model. The models used here are several types of YOLOv5, YOLOv6, and YOLOv7. Once trained, these models were applied to detect head movements, which were subsequently used to control the motor driver via an Arduino. The complete workflow can be seen in the flowchart in Fig. 2

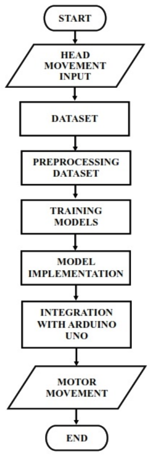


Fig. 2. The workflow of the control motor of Smart Wheelchair by head movement.

I. Hardware Configuration

Fig. 3 shows connection between the hardwares used. The black arrow shows the power connection between items, where the red arrows show both power and data. From the image, NUC plays the role of the processor, where the main process is done here. This is where the object detection model runs. It is then connected to the arduino and camera. The Arduino then sends PWM signals to the motor driver, which in turn controls the motor's movement based on the received signals. The driver then moves the motor based on the signal received. Power source used in the circuit is from the battery, which is connected to the switch and power indicator.

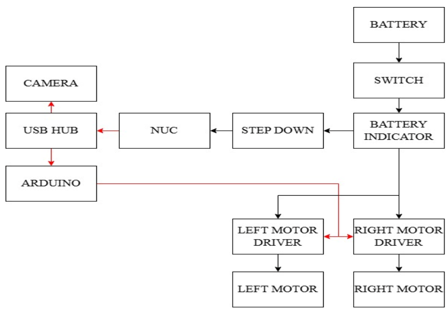


Fig. 3. The connection between the hardware used.

In order to get the wheelchair moving in the desired direction, both left and right motors are set. The motors will turn on and off based on the detection result. The details of the setting of the motor can be seen at Table V.

TABLE V. MOTORS MOVEMENT

Arah gerak kepala	Arah gerak kursi roda	Motor kanan	Motor kiri
Depan	Maju	On	On
Bawah	Berhenti	Off	Off
Kanan	Kanan	Off	On
Kiri	Kiri	On	Off

Based on Table V, when the system detects the head facing forward, the wheelchair will move forward by turning on both right and left motors. If the system detects the head facing downward, then the wheelchair will stop, by turning off both motors. When the system detects the head tilted to the right, then the wheelchair will turn right. This is done by turning on the left motor and turning off the right motor. On the other hand, if the system detects the head tilted to the left, the wheelchair will also turn left by turning on the right motor and turning off the left motor.

J. Design

Fig. 4. Prototype of smart wheelchair.

Fig. 4 shows the wheelchair's design with the dimension of 83 cm×49 cm×108 cm. There's a 38 cm×40 cm×6 cm storage box under the wheelchair, functioning to store electronic components and the circuit. Inside it, NUC, Arduino Uno, motor driver and cables can be found.

IV. RESULT AND DISCUSSION

A. Testing on the Dataset

When the model has trained, then it can be tested. Here, the model tested on 448 images, which is 20% from the whole dataset. The testing is done to compare the model speed when it's detecting the object in the testing images set. Here, we compare the detection results from several types of YOLOv5, YOLOv6, and YOLOv7. To make a balance comparison, all the models were trained on the same conditions, which is 60 epochs with 0.8 confidence level. The detailed results can be seen in Table VI for YOLOv5, Table VII for YOLOv6, and Table VIII for YOLOv7.

TABLE VI. RESULT FROM YOLOV5

Jenis	Average inference time	Average NMS time
YOLOv5 N	3.9 ms	3.2 ms
YOLOv5 S	7.0 ms	2.2 ms
YOLOv5 M	18.5 ms	2.1 ms
YOLOv5 L	29.3 ms	2.1 ms

Based on Table VI, YOLOv5N has the smallest average inference time, that is 3.9 ms. Yet for the average NMS time, both YOLOv5M and YOLOv5L are the smallest, that is 2.1 ms.

TABLE VII. RESULT FROM YOLOV6

Jenis	Average inference time	Average NMS time
YOLOv6 N	2.54 ms	2.55 ms
YOLOv6 S	4.23 ms	2.74 ms
YOLOv6 M	9.50 ms	2.51 ms
YOLOv6 L	15.37 ms	1.76 ms

From Table VII, YOLOv6N has the smallest inference time, 2.54 ms and the biggest is YOLOv6L, with the speed of 15.37 ms. For the Average NMS time, YOLOv6L has the fastest speed, 1.76 ms and YOLOv6S is the slowest, 2.74 ms.

TABLE VIII. RESULT FROM YOLOV7

Jenis	Average inference time	Average NMS time
YOLOv7	14.8 ms	1.6 ms
YOLOv7-X	15.5 ms	1.7 ms
YOLOv7-W6	14.8 ms	1.7 ms
YOLOv7-E6	13.2 ms	1.9 ms
YOLOv7-D6	14.1 ms	1.8 ms
YOLOv7-E6E	15.6 ms	1.7 ms

Table VIII shows that YOLOv7-E6 has the smallest inference time, with the speed of 13.2 ms and the longest is YOLOv7-E6E, that is 15.6 ms. For the NMS time, the speed is almost the same for the six variants, yet the fastest is YOLOv7 with the speed of 1.6 ms.

B. Model Evaluation

Besides testing the model based on its performance in detecting objects in images, evaluations were also done for several parameters. The parameters used to evaluate the model are P@.5iou, R@.5iou, F1@.5iou, mAP@.5, and mAP@.5:.95. The complete result of the evaluation can be seen in Tables VII–IX.

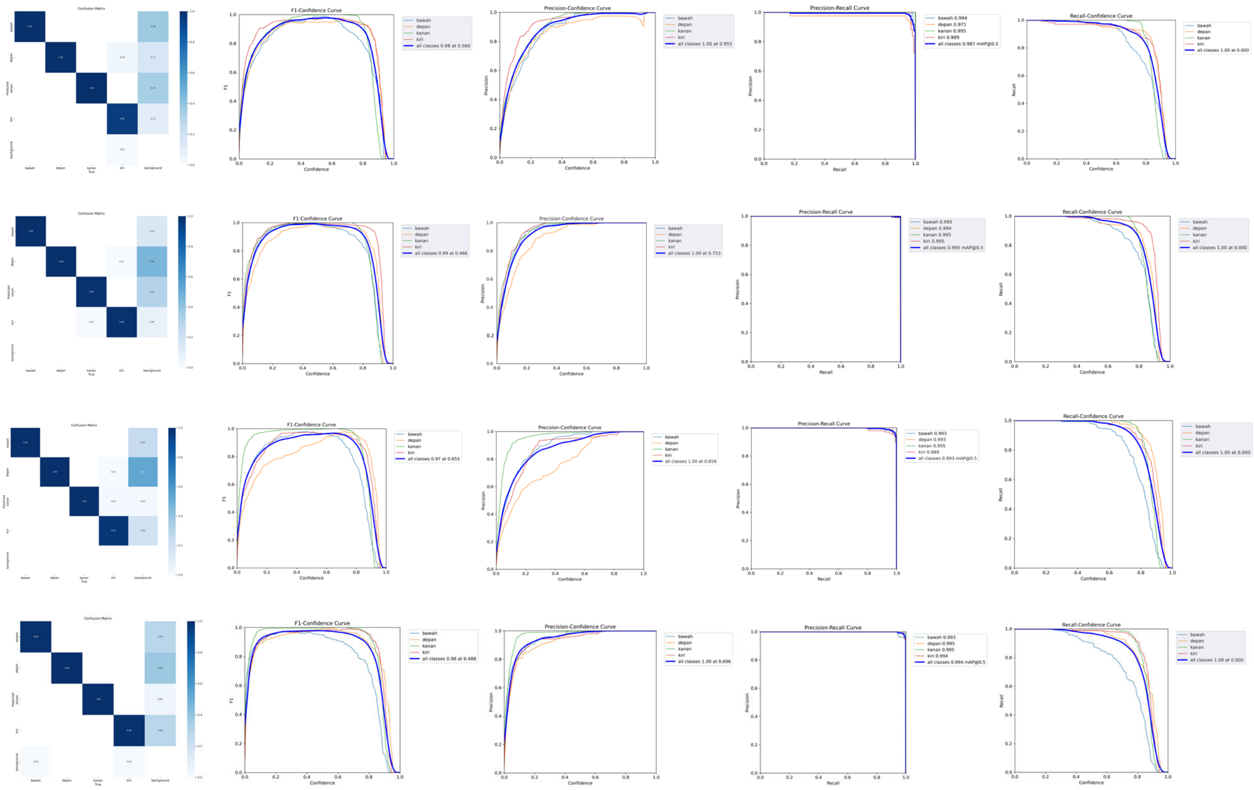
TABLE IX. EVALUATION FROM YOLOV5

Types	P	R	mAP50	mAP50-95
YOLOv5 N	0.965	0.97	0.993	0.703
YOLOv5 S	0.984	0.972	0.994	0.755
YOLOv5 M	0.99	0.94	0.995	0.772
YOLOv5 L	0.984	0.971	0.997	0.761

From Table IX, YOLOvM has the highest precision, that is 0.99 and the lowest is YOLOv5N, 0.965. For the recall, YOLOv5S has the highest recall with the score of

0.972 and YOLOv5M is the smallest, 0.94. YOLOv5M has the highest score of mean average precision from 50 to 95 (mAP50-95), with a score of 0.995. It also has the highest score for mean average precision at 50 (mAP50), 0.772.

Fig. 5 also shows the confusion matrix and the evaluation graph for each variation of YOLOv5.



(a) (b) (c) (d) (e)

Fig. 5. From columns (a) confusion matrix, (b) f1, (c) precision, (d) precision-recall, (e) recall, and from top rows YOLOv5N, YOLOv5S, YOLOv5M, YOLOv5L.

TABLE X. EVALUATION FROM YOLOV5

Types	P@.5iou	R@.5iou	F1@.5iou	mAP@.5	mAP@.5:.95
YOLOv6 N	0.998	0.99	0.994	0.997	0.807
YOLOv6 S	0.998	0.99	0.994	0.997	0.81
YOLOv6 M	0.998	0.99	0.994	0.998	0.802
YOLOv6 L	0.998	0.99	0.994	0.997	0.808

Table X shows that YOLOv6 has almost the same score for every parameter except for Mean Average Precision from 0.5 to 0.95 IoU (mAP@.5:.95). YOLOv6N has the value of 0.807, YOLOv6S 0.81, YOLOv6M 0.802, and YOLOv6L 0.808. YOLOv6M has a slightly different score for Mean Average Precision at 0.5 Intersection over Union (IoU) (mAP@.5), 0.998.

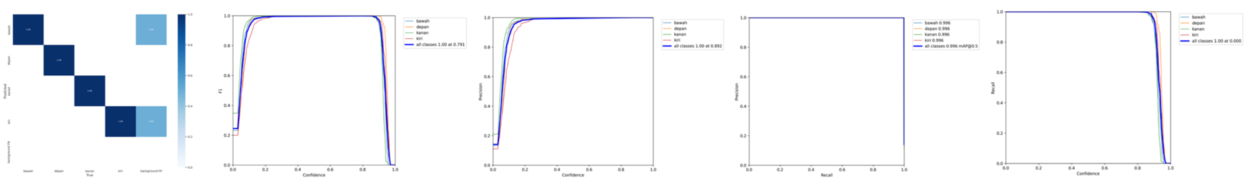
Fig. 6 also shows the confusion matrix and the evaluation graph for each variation of YOLOv6.

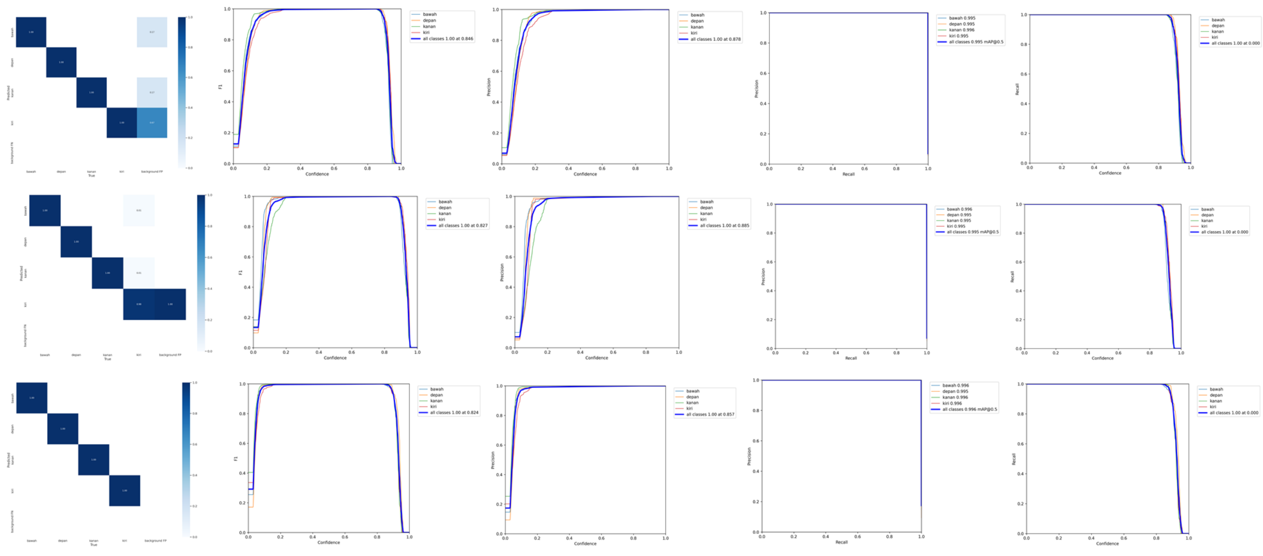
TABLE XI. EVALUATION FROM YOLOV7

Types	P	R	mAP@.5	mAP@.5:.95
YOLOv7	0.996	0.997	0.995	0.799
YOLOv7-X	0.997	0.998	0.995	0.801
YOLOv7-W6	0.991	0.993	0.995	0.797
YOLOv7-E6	0.994	0.992	0.995	0.785
YOLOv7-D6	0.996	0.996	0.995	0.789
YOLOv7-E6E	0.991	0.993	0.995	0.789

Table XI shows that YOLOv7-X has the highest precision, recall, and Mean Average Precision from 0.5 to 0.95 IoU (mAP@.5:.95). It has precision score of 0.997, recall score 0.998, and mAP@.5:.95 score of 0.801. For the Mean Average Precision at 0.5 Intersection over Union (IoU) (mAP@.5), all variance have the same score, 0.995.

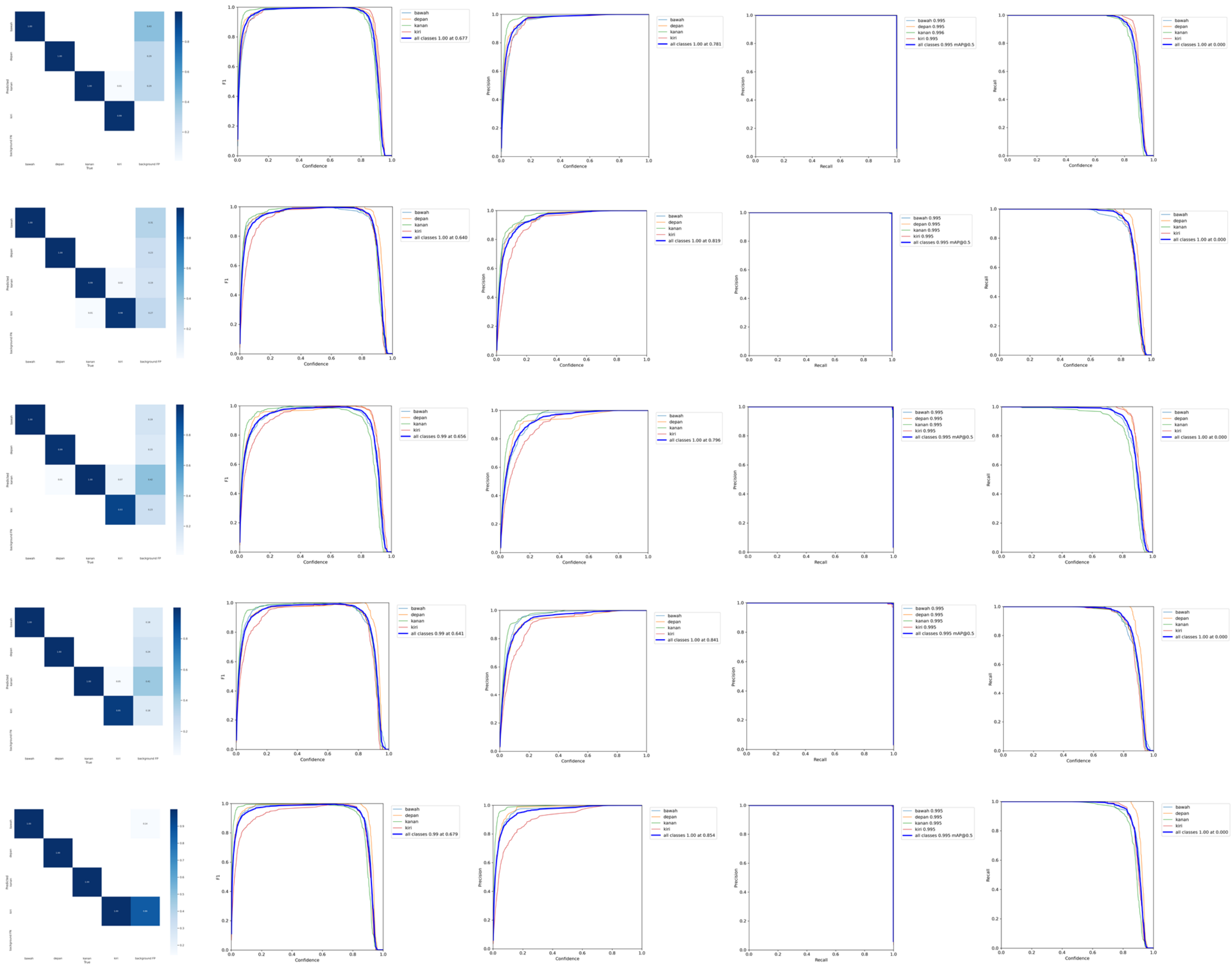
Fig. 7 also shows the confusion matrix and the evaluation graph for each variation of YOLOv7.

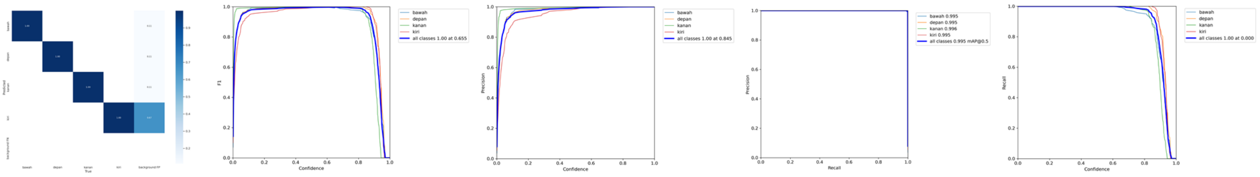




(a) (b) (c) (d) (e)

Fig. 6. From columns. (a) confusion matrix, (b) f1, (c) precision, (d) precision-recall, (e) recall and from top rows YOLOv6N, YOLOv6S, YOLOv6M, YOLOv6L.





(a) (b) (c) (d) (e)

Fig. 7. From columns (a) confusion matrix, (b) f1, (c) precision, (d) precision-recall, (e) recall, and from top rows YOLOv7, YOLOv7-X, YOLOv7-W6, YOLOv7-E6, YOLOv7-D6, YOLOv7-E6E.

C. Real-World Testing

To further validate the effectiveness of the smart wheelchair integrated with YOLOv6 in real-world usage scenarios, we conducted real-world testing involving three actual wheelchair users. The testing parameters include navigation accuracy, response time and user feasibility aspects such as safety, comfortable to use, and ease to use.

The testing was conducted in the main entrance and exit points of a university building with high foot traffic and various obstacles, including pillars, chairs, and automatic doors. Each participant was asked to move in the testing environment while avoiding existing obstacles using head movements.

During testing, we examined how the wheelchair responded to different head movements by analyzing motor activation. When the user turned their head right, the left motor stayed active while the right motor stopped, allowing the wheelchair to turn right. If the camera detects the user's movement to the left, the right motor will activate and the left motor will stop. Moving forward required both motors to run simultaneously, while tilting the head downward stopped both motors, bringing the wheelchair to a halt. To ensure safety, the system automatically switched to Free Mode and stopped moving if it couldn't reliably detect the user's head position, preventing unintended movements and improving overall navigation stability. The test results can be seen in Table XII.

TABLE XII. REAL-WORLD TESTING RESULTS

Participant	Navigation Accuracy (%)	Response Time (ms)	User Feasibility (1-4)		
			Safety	Comfortable	Ease to Use
P1	80%	331	4	3	3
P2	90%	487	4	4	2
P3	90%	384	4	3	3

Table XII shows the results of real-world testing, an average navigation accuracy of 86.67% indicating reliable head movement interpretation. The average response time of 401 ms ensuring real. Participants rate the feasibility positively, particulary for safety and comfort, though challenges were noted regarding ease of use and response time variability.

To further analyze the system's reliability, we evaluated the accuracy of the detected head direction by conducting tests using head direction image inputs with various test data samples, comparing the image processing results from the camera with the actual wheelchair movement in response to head movements. The results can be seen in Tables XIII–XVI.

TABLE XIII. INTEGRATION TEST RESULTS OF HEAD MOVEMENT DETECTION FOR RIGHTWARD MOVEMENT

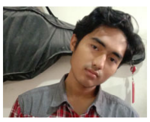
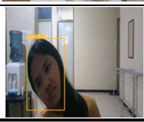
No	Input Image	Head Direction	Detection Result Image	Motor Activation		Integration
				Right Motor	Left Motor	
1		Right		Inactive	Active	Matches
2		Right		Inactive	Active	Matches
3		Right		Inactive	Active	Matches

Table XIII presents the results of the integration test for detecting head movement to the right. The system consistently identifies the head direction accurately and activates the motors as expected, with the left motor active and the right motor inactive, causing the wheelchair to move to the right as intended.

TABLE XIV. INTEGRATION TEST RESULTS OF HEAD MOVEMENT DETECTION FOR LEFTWARD MOVEMENT

No	Input Image	Head Direction	Detection Result Image	Motor Activation		Integration
				Right Motor	Left Motor	
1		Left		Active	In-active	Matches
2		Left		Active	In-active	Matches
3		Left		Active	In-active	Matches

The outcomes of the integration test for identifying leftward head movement are shown in Table XIV. Smooth and exact leftward movement was ensured by the system's accurate recognition of leftward head orientation, which triggered the right motor while deactivating the left motor.

TABLE XV. INTEGRATION TEST RESULTS OF HEAD MOVEMENT DETECTION FOR FORWARD MOVEMENT

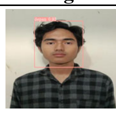
No	Input Image	Head Direction	Detection Result Image	Motor Activation		Integration
				Right Motor	Left Motor	
1		Front		Active	Active	Matches
2		Front		Active	Active	Matches
3		Front		Active	Active	Matches

Table XV outlines the integration test results for detecting forward head movement. The system consistently identified when the user faced forward and correctly activated both motors, ensuring the wheelchair moved straight ahead as intended with smooth and stable motion.

TABLE XVI. INTEGRATION TEST RESULTS OF HEAD MOVEMENT DETECTION FOR DOWNWARD MOVEMENT

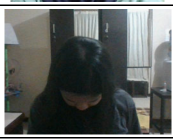
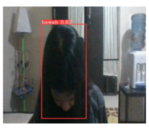
No	Input Image	Head Direction	Detection Result Image	Motor Activation		Integra-tion
				Right Motor	Left Motor	
1		Down		In-active	In-active	Matches
2		Down		In-active	In-active	Matches
3		Down		In-active	In-active	Matches

Table XVI shows the results of the integration test for downward head movement detection. When the system detected the user tilting their head downward, both motors were deactivated, bringing the wheelchair to a full stop.

D. Evaluation of Methodology

This research shows that the YOLO-based deep learning approach for head motion detection in wheelchair navigation has advantages over existing approaches (e.g., EEG and Piezoelectric based studies), but this study still has limitations for future development. Our studys show that the proposed methodology produces a wheelchair that is less intrusive and more user-friendly. Unlike previous methodologies that require electrodes or physical sensors attached to the user's body, our approach relies solely on computer vision, making it more practical for daily use.

From the analysis conducted, it was found that the YOLOv6N model had the fastest inference time (2.54 ms), outperforming other YOLO variations tested in this study. This finding supports the theory that lightweight deep learning models can be used for real-time assistive technology applications with minimal latency. Another promising finding is that the trained model achieved high detection accuracy, with a mean average precision (mAP@.5) value of 0.99 and a precision-recall score above 0.9. In line with previous studies on deep learning-based object detection, these results confirm that the methodology in this study is suitable for real-time mobility assistance systems. Furthermore, the proposed approach demonstrates scalability and adaptability, as the model can be retrained with additional datasets to accommodate different variations of user head movements. These findings are in line with previous studies that highlight the flexibility of deep learning models in assistive technology applications.

However, this study shows a dependence on environmental conditions such as lighting and background conditions. In the conducted tests, it was found that adaptive image processing techniques are needed to maintain system performance under various conditions. Additionally, further optimization is still needed for the high computational power requirements on low-power embedded devices. The results of this research still focus on the four directions head movement, unlike previous research that included more complex movement patterns, the model developed in this study has not yet been able to recognize diagonal head movements.

From real-world testing, it was observed that response times varied among participants, with an average latency of 401 ms. These results confirm the hypothesis that real-time performance can be achieved, but optimization is still needed to reduce variability between different users. Although the system shows an average navigation accuracy of 86.67%, further improvements in model adjustments and dataset expansion are needed to enhance the system's reliability.

V. CONCLUSION

The smart wheelchair is built using the YOLO model, a deep learning model based on Convolutional Neural Network (CNN). The model used to detect head movement direction, which will navigate the wheelchair's movement. There are four classes of head movements that can be detected: front, right, left and down. The front class is when the head is facing straight forward. While the left and right class, is when the head is tilted to left or right. Lastly, the down class is detected when the head is facing down.

After the model trained, it then tested and evaluated to see how well it is in detecting objects in images. From every variation of the three YOLO generations, YOLOv6N has the fastest inference time, that is 2.54 ms. For the NMS time, YOLOv7 has the fastest speed, that is 1.6 ms. For the evaluation result, there's no huge difference between each variation. All of the precision, recall and mAP@.5 of each variation are above 0.9. Yet, the difference can be seen for the mAP@.5:.95 where the highest score is 0.808 from YOLOv6L and the lowest is 0.703 from YOLOv5N.

Based on this study, real-world application was carried out using YOLOv6N, which has the fastest inference time, and this was also proven by testing in the real world where stable results were obtained. For further development and to provide more effective wheel function performance, a qualitative evaluation of the model needs to be conducted in more varied environmental conditions such as low-light conditions and occlusions. To overcome low-light issues, we acknowledge the potential benefits of multimodal sensor integration, such as adding infrared (IR) sensors to the vision-based YOLO model. Additionally, future improvement could focus on movement filtering techniques and adaptive thresholding to enhance stability in more dynamic environments.

CONFLICT OF INTEREST

The authors declare no conflict of interest.

AUTHOR CONTRIBUTIONS

F.U. supervised the project, provided guidance, and reviewed the manuscript and provided feedback. E.B. handled the electrical components, integrated the system, and contributed to writing and editing. R.N.F. managed requirements engineering and contributed to writing and editing. Y.C.A. worked on deep learning programming and contributed to writing and editing. A.Q. coordinated the ethics testing and contributed to the writing and editing of the manuscript. All authors have read and approved the final version of the manuscript.

ACKNOWLEDGMENT

The authors would like to express their gratitude to Universitas Brawijaya for their support and resources throughout this research grand Professor No. 02243/UN10.F1501/B/PT.01.05.1/ 2024.

REFERENCES

[1] Badan Pusat Statistik Kabupaten Kepulauan Talaud. (Desember 2021). Hari Disabilitas Nasional. [Online]. Available: https://talaudkab.bps.go.id/news/2021/12/03/74/hari-disabilitas-internasional.html

[2] C. Riyanti and N. C. Apsari, "Description of the self-actualization needs of people with physical disabilities through work," *Jurnal Pekerjaan Sosial*, vol. 3, no. 1, pp. 40–52, July 2020. (in Indonesia)

[3] GeeksforGeeks. (January, 2023). Advantages and disadvantages of deep learning. [Online]. Available: https://www.geeksforgeeks.org/advantages-and-disadvantages-of-deep-learning/

[4] N. Barazida. (July, 2022). YOLOv6: Next-generation object detection—Review and comparison. [Online]. Available: https://towardsdatascience.com/yolov6-next-generation-object-detection-review-and-comparison-c02e515dc45f

[5] L. Jiang *et al.*, "SmartRolling: A human–machine interface for wheelchair control using EEG and smart sensing techniques," *Information Processing & Management*, vol. 60, no. 3, 103262, May 2023. doi: 10.1016/j.ipm.2022.103262

[6] C. Bouyam and Y. Punsawad, "Human–machine interface-based wheelchair control using piezoelectric sensors based on face and tongue movements," *Heliyon*, vol. 8, no. 11, p. e11679, November 2022. doi: 10.1016/j.heliyon.2022.e11679

[7] Ultralytics. (November, 2023). Comprehensive guide to ultralytics YOLOv5. [Online]. Available: https://docs.ultralytics.com/yolov5/

[8] Y. Wang and L. Pan, "YOLOV5s-Face face detection algorithm," *2022 China Automation Congress (CAC)*, pp. 1107–1112, November 2022. doi: 10.1109/cac57257.2022.10054674

[9] K. Sudars *et al.*, "YOLOv5 deep neural network for quince and raspberry detection on RGB images," in *Proc. 2022 Workshop on Microwave Theory and Techniques in Wireless Communications (MTTW)*, October 2022, pp. 19–22. doi: https://doi.org/10.1109/mttw56973.2022.9942550

[10] M. Bie, Y. Liu, G. Li, J. Hong, and J. Li, "Real-time vehicle detection algorithm based on a lightweight You-Only-Look-Once (YOLOv5n-L) approach," *Expert Systems with Applications*, vol. 213, 119108, March 2023. doi: 10.1016/j.eswa.2022.119108

[11] S. Tan, G. Lu, Z. Jiang, and L. Huang, "Improved YOLOv5 network model and application in safety helmet detection," in *Proc. 2021 IEEE International Conference on Intelligence and Safety for Robotics (ISR)*, March 2021.

[12] G. Liu, Y. Hu, Z. Chen, J. Guo, and P. Ni, "Lightweight object detection algorithm for robots with improved YOLOv5," *Engineering Applications of Artificial Intelligence*, vol. 123, pp. 106217–106217, August 2023. doi: 10.1016/j.engappai.2023.106217

[13] S. Luo and J. Liu, "Research on car license plate recognition based on improved YOLOv5m and LPRNet," *IEEE Access*, vol. 10, pp. 93692–93700, January 2022. doi: https://doi.org/10.1109/access.2022.3203388

[14] N. Singh, P. Saini, O. Shubham, R. Awasthi, A. Bharti, and N. Kumar, "Improved YOLOv5l for vehicle detection: An application to estimating traffic density and identifying over speeding vehicles on highway scenes," *Multimedia Tools and Applications*, vol. 83, pp. 5277–5307, May 2023. doi: 10.1007/s11042-023-15520-9

[15] T. Davies. (Juny, 2022). MT-YOLOv6: A YOLO-inspired object detection model released. [Online]. Available: https://wandb.ai/telidavies/ml-news/reports/MT-YOLOv6-A-YOLO-Inspired-Object-Detection-Model-Released--VmlldzoyMjMzMzI5

[16] A. Nazir and M. A. Wani, "You Only Look Once—Object detection models: A review," in *Proc. 2023 10th International Conference on Computing for Sustainable Global Development (INDIACom)*, New Delhi, India, 2023, pp. 1088–1095.

[17] C. Li *et al.*, "YOLOv6: A single-stage object detection framework for industrial applications," arXiv preprint, arXiv:2209.02976, 2022. doi: 10.48550/arxiv.2209.02976

[18] Meituan. (September, 2023). YOLOv6. [Online]. Available: https://github.com/meituan/YOLOv6

[19] C.-Y. Wang, A. Bochkovskiy, and H.-Y. M. Liao, "YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors," in *Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 7464–7475. doi: 10.48550/arxiv.2207.02696

[20] Ultralytics. (November, 2023). YOLOv7: Trainable bag-of-freebies. [Online]. Available: https://docs.ultralytics.com/models/yolov7/

[21] M. Kachurova, M. Porjazoski, P. Latkoski, B. Popovski, and G. Stevanoski, "Comparative analysis of YOLOv7 models for use in perimeter protection of military areas," in *Proc. 2023 30th International Conference on Systems, Signals and Image Processing (IWSSIP)*, Ohrid, North Macedonia, 2023, pp. 1–5. doi: https://doi.org/10.1109/iwssip58668.2023.10180260

[22] Z. E. Baiat and Ş. Baydere, "Smart city traffic monitoring: YOLOv7 transfer learning approach for real-time vehicle detection," in *Proc. 2023 International Conference on Smart Applications, Communications and Networking (SmartNets)*, Istanbul, Turkiye, 2023, pp 1–6. doi: 10.1109/smartnets58706.2023.10216009

[23] X. Chen and Q. Xie, "Safety helmet-wearing detection system for manufacturing workshop based on improved YOLOv7," *Journal of Sensors*, vol. 2023, pp. 1–14, May 2023. doi: 10.1155/2023/7230463

[24] Z. Lin, Y. Wang, J. Zhang, and X. Chu, "DynamicDet: A unified dynamic architecture for object detection," in *Proc. 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Vancouver, BC, Canada, 2023, pp. 6282–6291. doi: 10.1109/cvpr52729.2023.00608

[25] K.-Y. Wong. (September, 2023). Official YOLOv7. [Online]. Available: https://github.com/WongKinYiu/yolov7

[26] R. Chauhan, K. K. Ghanshala, and R. C. Joshi, "Convolutional Neural Network (CNN) for image detection and recognition," in *Proc. 2018 First International Conference on Secure Cyber Computing and Communication (ICSCCC)*, Jalandhar, India, 2018, pp. 278–282. doi: 10.1109/ICSCCC.2018.8703316

[27] S. Ilahiyah and A. Nilogiri, "Implementation of Deep Learning in Identification of Plant Types Based on Leaf Images Using Convolutional Neural Network," *JUSTINDO (Jurnal Sistem dan Teknologi Informasi Indonesia*), vol. 3, no. 2, pp. 49–56, August 2018. doi: https://doi.org/10.32528/justindo.v3i2.2254 (in Indonesia)