IJRCS

International Journal of Robotics and Control Systems
Vol. 6, No. 2, 2026, pp. 1115-1138
ISSN 2775-2658
http://pubs2.ascee.org/index.php/ijrcs

ASCEE

Illumination-Robust Head Pose Estimation for Enhanced Smart Wheelchair Control

Fitri Utaminingrum [a,1,*], I Komang Somawirata [b,2], Rahma Nur Fitriyani [b,3], Amila Fadhila Rahmaniati [b,4]

[a] Faculty of Computer Science, Universitas Brawijaya, Malang, Indonesia
[b] Department of Electrical Engineering, Institut Teknologi Nasional Malang, Malang, Indonesia
[1] f3_ningrum@ub.ac.id; [2] kmgsomawirata@lecturer.itn.ac.id; [3] rahmanur24@student.ub.ac.id; [4] amilafr@student.ub.ac.id
[*] Corresponding Author

ARTICLE INFO

Article History
Received December 19, 2025
Revised February 05, 2026
Accepted April 11, 2026

Keywords
AutoContrast;
DeepISP;
Head Pose;
Histogram Equalization;
Smart Wheelchair

ABSTRACT

Smart wheelchairs based on head pose estimation offer an alternative solution for individuals with limited arm and leg function to aid mobility. However, their performance is significantly affected by illumination conditions, particularly in low-light environments. Most previous studies lack real-world implementation and have only been evaluated in laboratory settings. To address these limitations, this research investigates the implementation of image preprocessing techniques to enhance the robustness of head-controlled wheelchair systems under different lighting conditions. This study presents a comparative analysis of three image preprocessing techniques: Histogram Equalization, AutoContrast, and DeepISP, evaluated on an actual smart wheelchair prototype. Experimental results show that DeepISP consistently achieves the highest accuracy, reaching 0.942 in normal-light and 0.933 in low-light environments, resulting in more stable and reliable head movement detection. Although DeepISP requires slightly higher processing time (34.82 ms), the latency remains within real-time constraints, representing an acceptable trade-off between accuracy and responsiveness. These findings highlight the importance of illumination enhancement in improving usability, safety, and reliability of vision-based assistive wheelchair systems across diverse lighting conditions.

1. Introduction

According to data released by the WHO (World Health Organization), an estimated 1.3 billion individuals worldwide live with disabilities [1]. These disabilities cover a wide range of conditions, including physical disabilities. For a condition to be classified as a physical disability, it must significantly impair an individual's ability to perform physical activities [2]. Individuals with physical disabilities are often diagnosed with motor impairments, which can lead to partial or complete loss of mobility in the limbs, affecting arms, legs, or both [3]. Individuals with physical disabilities frequently have limits in independent mobility, and they typically rely on other family members to travel throughout their territory or residence. Although physical activity is essential for maintaining overall health, participation levels among people with disabilities remain considerably low, primarily due to mobility challenges [4].

Numerous factors contribute to the emergence of physical disabilities, one of which is an accident. Traumatic accidents, especially those involving road traffic, are a primary etiology of amputation. As evidenced by Asirdizer et al. (2022), a substantial 58.8% of traumatic amputations are attributed to vehicular accidents [5]. The subsequent loss of a limb frequently necessitates the utilization of assistive technologies, with wheelchairs being a prevalent choice to facilitate mobility.

A wheelchair is an assistive device that can benefit patients, especially those with physical disabilities, by improving their health, well-being, participation, and inclusion [6]. The World Health Organization (WHO) has estimated that approximately 10% of the global population has disabilities, with 10% of this disabled population requiring wheelchairs for mobility [7]. Wheelchairs are typically categorized into two primary types: manual wheelchairs and electric wheelchairs. The operation of both manual and electric wheelchairs relies on manual control, necessitating hand assistance. However, this poses a considerable challenge for individuals contending with multiple physical disabilities, particularly those encountering difficulty or inability to maneuver their arms and legs [8], [9].

Previous studies have been undertaken to address alternative navigation for wheelchairs [10]. Joly et al. (2023) proposed a voice-controlled wheelchair that allows physically disabled people who are unable to regulate their movements, particularly with their hands, to drive the wheelchair independently [11]. The user can operate this system by using basic voice commands. A voice recognition module connected to an Arduino is used to recognize speech. The wheelchair's basic components are a voice recognition module and a motor system controlled by a microcontroller. Iskanderani et al. (2021) [12] and Venkatakrishnan et al. (2024) [13] also proposed a voice controlled smart wheelchair to cover for joystick as navigation. They both use a voice recognition module controlled by a microcontroller. A wheelchair controllable through voice was also developed by Hariharan S. et al. (2025) for physical impaired individuals [14]. This study incorporates microphone array and noise-cancellation algorithms into speech recognition technology, allowing users to control wheelchair's movements with simple voice commands to ensure accurate voice recognition in a variety of environments, including noisy ones.

Although those studies effectively facilitated mobility for individuals with multiple physical disabilities, they exhibited limitations. The reliance on sound as the primary control mechanism renders the wheelchairs susceptible to fragility and sensitivity to ambient noise, thereby restricting wheelchair usage in more quiet environments. To mitigate these limitations, an alternative control system utilizing head movements is proposed, offering users greater ease and precision in wheelchair control. The digital image-based approach proved effective in precisely detecting small changes in objects [15]. This indicates that similar visual technology has the potential to be effectively adapted for smart wheelchair navigation systems [16], particularly head-pose-based, supporting improved user experience.

Numerous studies have explored the application of vision-based head-pose estimation for wheelchair control, notably through two significant approaches. One study developed a smart wheelchair that employs the Haar Cascade algorithm, utilizing facial and nasal recognition to interpret four distinct movements [17]. "Look down" when the nose approaches the bottom of the face bounding box, "Look up" when the nose nears the center, "Turn right" when the nose is close to the right, and "Turn left" when the nose is proximate to the left. Another study introduced an electric wheelchair controlled via head movements using the YOLOv3 algorithm, categorizing movements into five types: backward when lowering the head, forward when raising the head, neutral when looking forward, left when tilting to the left, and right when tilting to the right [18].

Fig. 1 describes an overview of the smart wheelchair navigation system that employs camera-based head pose estimation. This navigation method is divided into three primary steps. The first stage, illustrated in Fig. 1 (a), involves head pose detection using a camera positioned facing the user. Fig. 1 (b) shows how the direction of this head movement is used to regulate the signal to the motor

driver, which controls the movement of the right and left wheels of the wheelchair. Furthermore, Fig. 1 (c) depicts the movement of the wheelchair as a result of wheel movement.

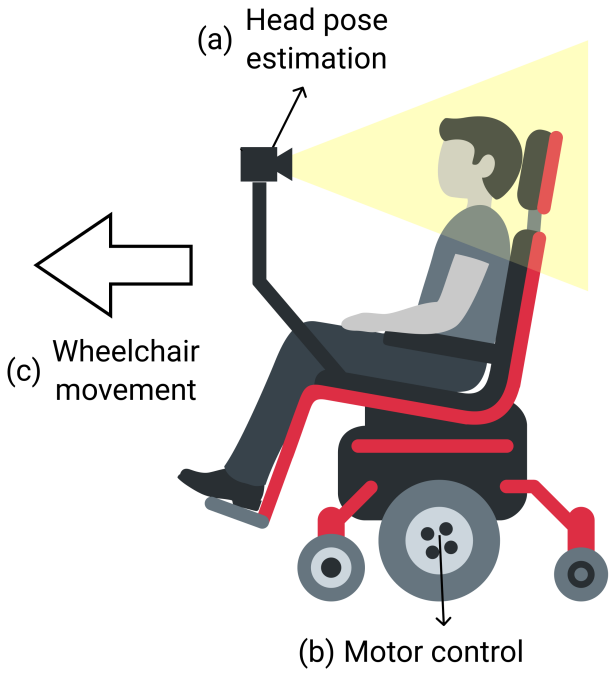


Fig. 1. Overview of vision-based head pose estimation for navigation system in smart wheelchairs

While previous studies demonstrate significant advancements in enabling vision-based head pose estimation for wheelchair control for individuals with disabilities, they share a common limitation. The requirement for adequate lighting conditions makes these systems less optimal when operating in low-light environments. Low light conditions result in low contrast, high noise, and the loss of critical feature information [19], thus decreasing the accuracy of head pose estimation. This poses significant challenges for users, especially in real-world situations where lighting cannot always be controlled. In dark or low-light conditions, a system often experiences decreased performance, characterized by increased errors in object detection and recognition [20]. It can be dangerous for smart wheelchair users, as detection errors in low-light conditions can potentially cause the system to misunderstand directions. This results in inaccurate navigation and can increase the risk of accidents or incidents that could be detrimental to the user [21]. Furthermore, this difficulty is consistent with studies that emphasize how crucial efficient multimedia processing is to guaranteeing precise visual interpretation under challenging circumstances [22].

Low-light object detection is crucial for security, surveillance, and safety in various applications, including autonomous vehicles, search and rescue operations [23]. In our previous work [21], we developed a head-motion-based intelligent wheelchair control system that used RetinexNet to improve visual visibility in low-light circumstances. Using EfficientNetv2, the system achieved high classification accuracy in both normal and low-light environments. However, this approach was limited to a single illumination enhancement technique and only tested in an offline laboratory setting. In a follow-up study [24], we investigated an attention mechanism for dealing with different illumination situations. Although the system performed admirably in the laboratory environments, it remained unstable in the real world, with the model prone to overfitting in low-light conditions.

To address these limitations, this study focuses on improving the robustness of the head pose-based wheelchair navigation system in various lighting conditions, including low-light. We compare three illumination enhancement algorithms, Histogram Equalization, AutoContrast, and DeepISP,

to the YOLOv8 detection approach. In order to enable a thorough comparison in terms of visual quality, accuracy, and inference speed in a real-time head-controlled wheelchair system, those three enhancement techniques were chosen to represent global enhancement (Histogram Equalization), adaptive enhancement (AutoContrast), and learning-based enhancement (DeepISP).

Unlike previous studies that rely on a single approach and presume sufficient lighting, this study provides a thorough evaluation of enhancement strategies on a real-world hardware implementation using an in-house dataset gathered in both normal and low-light conditions. The novelty of this study lies in the systematic comparison of enhancement methods for real-time head gesture navigation and their implementation on a real smart wheelchair, enabling its robustness in various lighting conditions. The contributions of this paper are summarized as follows:

- We propose an illumination-robust pipeline for head-pose-based wheelchair navigation in multi-lighting conditions.
- We present a comparative analysis of three image preprocessing techniques (AutoContrast, Histogram Equalization, and DeepISP) in a YOLOv8 detection framework.
- We demonstrate testing on real-devices used in smart wheelchairs.

The objective of this study is to investigate the feasibility of expanding the operational capabilities of head-controlled wheelchairs beyond well-lit environments. By implementing image preprocessing techniques, we aim to develop a system that can effectively process images captured in diverse lighting conditions, thereby increasing the usability of the smart wheelchair. We employ the AutoContrast function of Pillow, Histogram Equalization, or the DeepISP algorithm, so adjustments can be made to the input image fed into the system. It is used to enhance the brightness of the input image, enabling the system to effectively process and detect captured head movements. To detect head movements, the system will utilize one of the deep learning models, namely YOLOv8. Deep learning is utilized in a wide range of applications such as healthcare, security, change detection, and many more [25]. YOLOv8 was specifically chosen due to its superior accuracy when compared to other YOLO models [26], [27]. Furthermore, the proposed system employs the N variation of the YOLOv8 model. The selection of YOLOv8 variation N (nano) is motivated by its lightweight nature in comparison to other variations. This approach aims to facilitate enhanced mobility for individuals with multiple physical disabilities not only in normal lighting environments but also in conditions with minimal lighting.

2. Method

The purpose of this method is to improve the independence and mobility of people with disabilities by enabling wheelchair control through head pose estimation. The system functions by capturing the user's head pose through a camera. Using a camera, the device records the user's head motions: straight ahead, to the right, to the left, or downward. These captured images are then processed to identify the direction of the user's head pose. Subsequently, the system transmits commands to the wheelchair's motors, enabling it to move in accordance with the detected head direction. The wheelchair is then able to travel in the direction of the identified head pose because the system has sent commands to its motors. Fig. 2 presents the end-to-end data flow of the proposed system, highlighting the sequence from data acquisition to wheelchair motor control.

To achieve this, the system employs advanced computer vision techniques to analyze the captured images. The combination of robotics with computer vision is a crucial technology driver in the automation of tasks that need strong perception, dependable planning, and precise actuation in unstructured settings [28]. The captured images undergo image preprocessing, including resizing and illumination enhancement, to improve visual quality before being fed into the deep learning model. Deep learning algorithm, specifically the YOLOv8-N model, is used to accurately detect and classify the user's head position. In computer vision and artificial intelligence, deep learning has

produced numerous compelling outcomes in addressing issues like object identification, recognition, classification, image segmentation, etc [29].

Fig. 2. End-to-end workflow of the proposed head-controlled wheelchair system

Following the end-to-end overview, the detailed workflow of each processing stage is described as follows. Within the input stage, the webcam serves dual functions by collecting data for both the training and inference phases. The system workflow consists of four main stages: preprocessing, training, detection, and motor control. In the preprocessing stage, there exist two sequential processes: image collection and preprocessing function application. The preprocessed images are used to train the YOLOv8-N model, improving model's ability to accurately detect and respond to distinct head movements, contributing to the overall efficacy of the system. Separate YOLOv8-N models are trained independently for each preprocessing technique. As a result, three trained models are produced, corresponding to the applied preprocessing techniques: AutoContrast, DeepISP, and Histogram Equalization.

In the inference stage, the trained models are subsequently integrated into the Jetson Nano. The webcam captures real-time head pose images, which are processed by the Jetson Nano to generate detection classes for each head movement. Each detection class is associated with a specific command, that issues a command to set the Pulse Width Modulation (PWM) value. This adjustment controls the movement of the wheels on the wheelchair, facilitating the translation of the user's head pose into specific and corresponding wheelchair actions. This closed-loop system, from image capture to model inference and subsequent motor control, ensures a responsive and adaptive mechanism for translating user head pose into meaningful wheelchair movement. The detailed workflow of the proposed method is illustrated in Fig. 3. During real-time operation, the system performs image acquisition, preprocessing, YOLOv8-N inference, and motor command generation in a continuous pipeline, allowing responsive wheelchair control with low latency.

2.1. Hardware Components

System is constructed utilizing a variety of hardware components, focused on the control system. The main components include Arduino UNO, motor driver, Jetson Nano, a webcam, and additional peripherals. Fig. 4 shows the hardware design of the head-pose-based smart wheelchair control system. Each device within the system performs a specific task to facilitate the overall functionality. The webcam plays a dual role in the system, firstly, to gather images for creating the dataset, and secondly, to capture users' head pose for inference during real-time usage. We use a Logitech HD C270 webcam as the image acquisition device, which provides a maximum resolution of 720p with a frame rate of up to 30 FPS. The Jetson Nano undertakes the image processing responsibilities, because it has CUDA which can be used for data processing. Following image processing, the results are transmitted to Arduino UNO, which, in turn, issues commands to the motor driver.

Communication between the Jetson Nano and Arduino UNO is established via serial communication (UART), where the Jetson Nano sends head pose classification results as control commands. The Arduino UNO interfaces with the IBT-2 H-Bridge motor driver through GPIO pins, generating Pulse Width Modulation (PWM) signals to regulate motor speed and direction. Direction control is handled through digital GPIO signals, while PWM signals are used to control the rotational speed of each motor independently.

To operate the wheel, the IBT-2 H-Bridge motor driver receives a PWM signal from the Arduino

UNO. The IBT-2 H-Bridge motor driver was chosen because of its function to control PWM, to interpret the commands from Arduino UNO and executes the corresponding actions on the wheelchair's wheels. The motor driver acts as a motor controller, regulating motor voltage and rotational direction. The system integrates two motor drivers to move the wheelchair, one on the right and one on the left. The Arduino UNO ensures consistent PWM values across both drivers. The system encompasses four potential movements: move forward (head looking forward), stop (head looking down), turn right (head tilted to the right), and turn left (head tilted to the left).

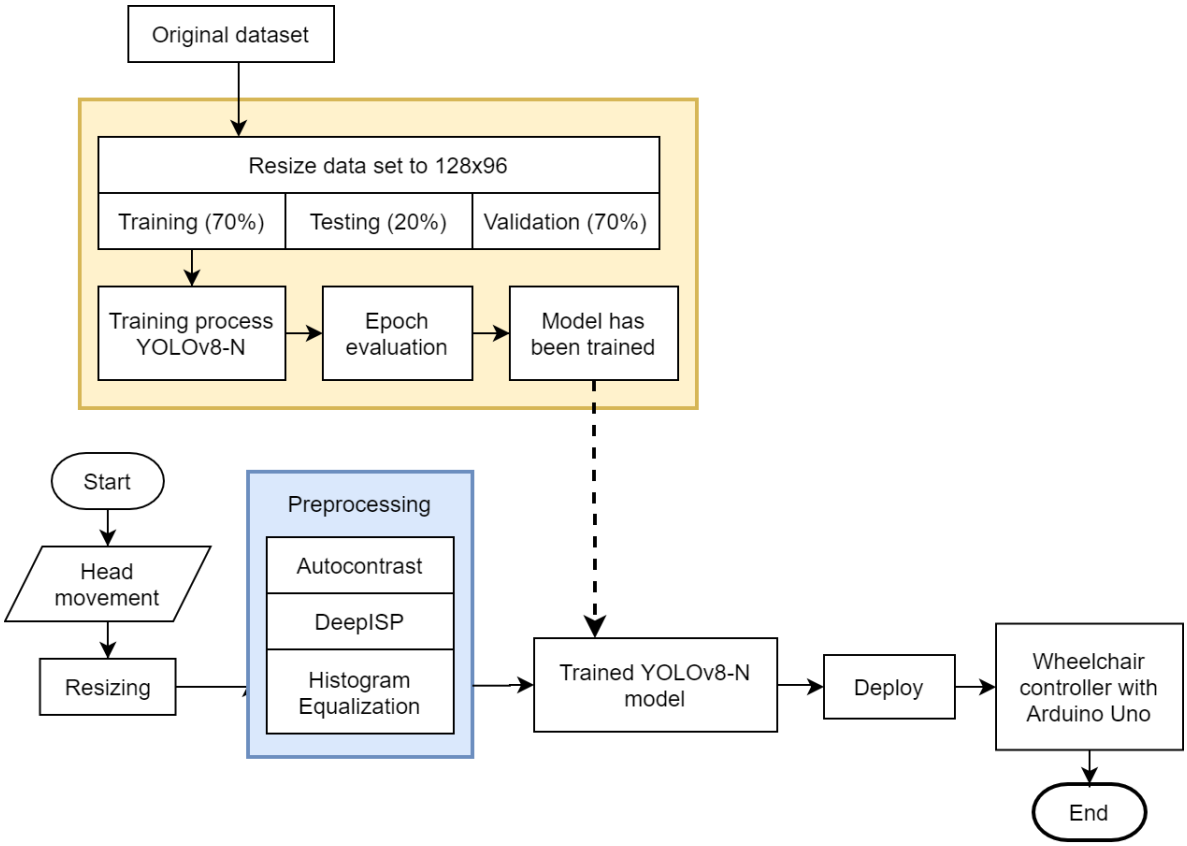


Fig. 3. Detailed workflow of proposed method

Table 1 displays the wheelchair movement commands determined by the operation of the left and right motor drivers. Each class or direction of head pose represents a wheelchair navigation command that is dependent on the operation of the right and left motor drivers. The wheel is driven by providing a PWM value to the motor driver. The right and left motor drivers control the right and left wheels of the smart wheelchair respectively, allowing it to move forward, stop, turn right, or turn left. If the system identifies a "front" class, both motors are triggered simultaneously with a PWM value of 40 on each motor driver, resulting in forward motion. When a "down" class is detected, both motors are turned off with a zero PWM value, causing the wheelchair to stop. For "right" class, the right motor is turned off (PWM = 0), but the left motor remains active (PWM = 40), causing the wheelchair turns to the right. In the "left" class, the right motor is activated (PWM = 40) while the left motor is switched off (PWM = 0), resulting in movement to the left.

2.2. Data Preparation

Dataset plays a crucial role in the training and evaluating of machine learning (ML) models [30]. A high-quality and well-structured dataset is essential for training machine learning models to accurately interpret and respond to user inputs, such as head pose.

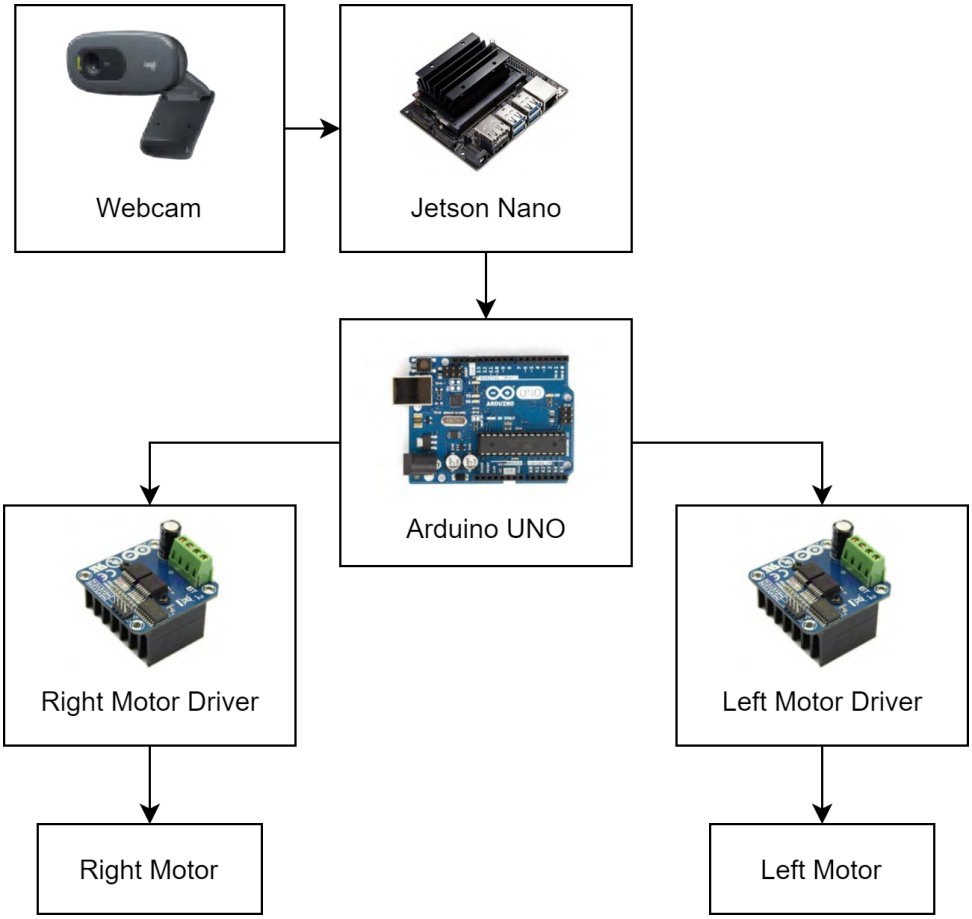


Fig. 4. Control system hardware design

Table 1. Operation of Left and Right Motors

Head Pose	Command on wheelchair	PWM Value	
		Left Motor	Right Motor
Looking forward	Move forward	40 (ON)	40 (ON)
Looking down	Stop	0 (OFF)	0 (OFF)
Tilted to the right	Turn right	40 (ON)	0 (OFF)
Tilted to the left	Turn left	0 (OFF)	40 (ON)

By capturing a diverse range of images under different lighting conditions, the dataset ensures that the system is robust and adaptable to real-world environments. Additionally, the variety of movement classes allows the system to effectively distinguish between different directional commands, making it a key component in enhancing the overall functionality and reliability of the smart wheelchair. We collected the dataset using a camera mounted on the wheelchair, positioned in front of the user at a distance of approximately 60 to 80 cm. This setup ensures the system can effectively capture head pose within a practical range during implementation process.

Subsequently, the processed datasets were uploaded to the Roboflow platform for the annotation process. Following annotation, the images were resized from their original dimensions of 640 $\times$ 480 pixels to 128 $\times$ 96 pixels. The dataset then underwent a splitting process, dividing it into three subsets: 70% for training, 10% for validating, and 20% for testing. The training dataset is utilized to establish and optimize model parameters, while the test dataset is specifically reserved to assess and evaluate the performance of the trained model. The overall workflow of dataset preparation process is illustrated in Fig. 5.

A dataset containing 2,666 images was used in this study, with the images classified into four distinct categories: front, down, left, and right. Sample images from each class are shown in Fig. 6 and Fig. 7. These images were captured under two different lighting conditions: normal-light (bright) environments, as shown in Fig. 6, and low-light (dark) environments, as shown in Fig. 7.

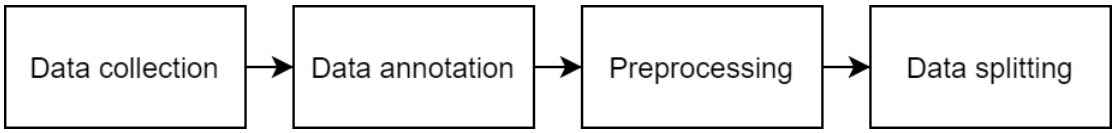
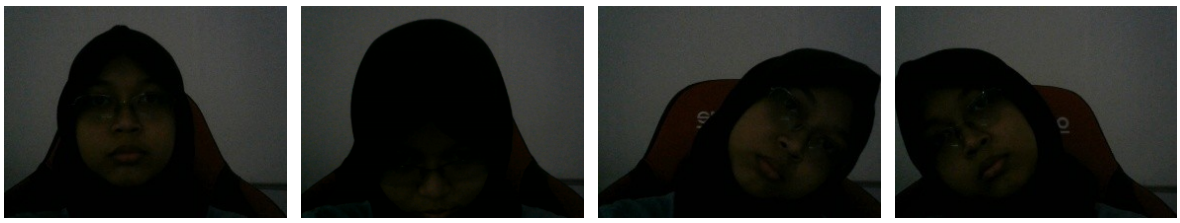


Fig. 5. Data preparation workflow.

These figures provide a visual representation of the sample images from each class in both light and dark conditions. The inclusion of multiple lighting conditions and distinct head pose classes increases dataset diversity, enabling the model to learn robust visual features that generalize well across varying illumination and real-world operating scenarios. Notably, the dataset images possess dimensions of 640 $\times$ 480 pixels.

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

Fig. 6. Sample images from the dataset of each class in normal-light condition: (a) front; (b) down; (c) right; and (d) left



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

Fig. 7. Sample images from the dataset of each class in low-light condition: (a) front; (b) down; (c) right; and (d) left

To ensure consistency with the four head pose classes employed in this study (front, down, right, and left), approximate angle thresholds were established during the dataset collection procedure, as shown in Table 2. Because the subject must constantly face the camera, yaw rotation is not used in this work. Instead, pitch represents vertical head motions for the down class, whereas roll represents lateral head tilting (left/right). These angle references were used only as guidance during data collection to ensure consistent head pose execution for each class across the samples. The final control performance is therefore determined by the predicted head pose classes, ensuring responsive and robust wheelchair operation under varying illumination conditions.

2.3. Preprocessing

Data preprocessing is a crucial step aimed at improving the quality of the data that will be used [31]. In the preprocessing stage in our proposed method, each image captured by the camera is first resized from its original dimensions of 300300 pixels to 128 $\times$ 96 pixels. This resizing step is crucial for reducing computational complexity and ensuring faster processing without significantly compromising image quality. In the preprocessing stage, each image captured by the camera is first analyzed to determine whether it was taken in a low-light or normal-light environment. This is done by calculating the average pixel intensity of the image. If the average pixel value indicates that the image was taken in a low-light condition, an image enhancement process is applied to improve its quality before further analysis.

Table 2. Head pose angle thresholds for wheelchair control

Class	Pitch (°)	Roll (°)	Description
Front	$-10 \leq$ Pitch ≤ 10	$-10 \leq$ Roll ≤ 10	Neutral head position
Down	Pitch < -15	-	Head nodding downward
Right	-	Roll > 15	Head tilted to the right
Left	-	Roll < -15	Head tilted to the left

Three different enhancement techniques are utilized for this purpose: Histogram Equalization, AutoContrast, and DeepISP. Histogram Equalization is utilized as a lightweight baseline for global contrast enhancement, because of its simplicity and cheap computational cost [32], [33]. This method redistributes the intensity values of the image to enhance contrast, particularly in darker regions, ensuring a more uniform appearance [34], [35]. AutoContrast is a more adaptive traditional method for adjusting pixel intensity distributions without requiring considerable processing overhead. It automatically improves image contrast by expanding the pixel intensity range, making features more visible in low-contrast areas [36]. DeepISP (Deep Image Signal Processing) is a learning-based enhancement technique that can restore small features and provide visually consistent illumination in low-light circumstances [37], [38]. DeepISP is a deep learning-based method that enhances image quality by mimicking the processing capabilities of professional cameras, making it particularly effective in challenging lighting conditions.

For real-time head-pose estimation, the three enhancement techniques enable a fair trade-off analysis between image quality, detection accuracy, and processing latency. These techniques fall into the global, adaptive, and deep learning categories respectively. By applying these techniques, the system ensures that the images are of sufficient quality for accurate interpretation and analysis, regardless of the lighting conditions at the time of capture.

2.3.1. Histogram Equalization

Histogram Equalization (HE) is a straightforward and effective method for enhancing contrast and improving image quality [39], [40]. It utilizes a cumulative distribution function (CDF) for pixel output intensity mapping [41]. This method involves two primary stages: calculating the CDF value from the image and mapping pixel values to a new range by multiplying the CDF value by the maximum pixel value and rounding to the nearest integer [42]. According to Chen et al. (2023), the calculation for this method can be expressed in (1) [43]. Equation (1) is used to calculate the probability of the histogram value $h(k)$ appearing in the range k from 0 to $L-1$, where L is the number of intensity levels in the input image with size $M \times N$.

$$p(k) = \frac{h(k)}{M \times N}, \quad \text{for } k = 0, 1, 2, \ldots, L-1 \tag{1}$$

Equation (2) is used to calculate the $CDF(j)$ value.

$$CDF(j) = \sum_{k=0}^{j} p(k), \quad \text{for } k = 0, 1, \ldots, j \text{ and } 0, 1, \ldots, L-1 \tag{2}$$

Equation (3) is used to change the value at each intensity level.

$$s = round((L-1) \times CDF(I)) \tag{3}$$

2.3.2. AutoContrast

AutoContrast is one of the functions contained in the ImageOps module from the PIL (Pillow) library, used to maximize the contrast of an image. AutoContrast stretches the image histogram to cover the full range of possible pixel values, enhancing contrast but potentially clipping highlights or shadows [36]. This function operates by first calculating a histogram from the image, subsequently, a cutoff percentage of the lightest and darkest pixels from the histogram are removed, which are then mapped to black (0) and white (255), respectively [44]. The formula of AutoContrast is shown in (4).

$$I'(x, y) = \frac{I(x, y) - I_{min}}{I_{max} - I_{min}} \times (L-1) \tag{4}$$

Equation (4) will calculate the amount of low-end pixels to be removed. $I'(x, y)$ represents the new pixel value at coordinates (x, y) after AutoContrast has been applied, while $I(x, y)$ denotes the original pixel value at the same coordinates. To apply AutoContrast, firstly, determine minimum and maximum values by identifying the minimum I_{min} and maximum I_{max} pixel intensity values in the image. The second is rescale pixel values using the formula in (4) to rescale each pixel intensity $I(x, y)$, so that it spans the entire range [0, $L-1$], typically [0, 255] for 8-bit images. The last is apply the new contrast levels by generate the output image by adjusting the pixel intensities according to the expanded range.

2.3.3. DeepISP

DeepISP (Deep Image Signal Processor) represents a method employed for image processing, leveraging a deep learning framework to emulate the complex processing pipeline of a digital camera, producing more natural and visually appealing results [45]. This approach offers the advantage of implicitly learning the statistical properties of natural images and simultaneously addressing multiple tasks, thereby potentially reducing the overall computational burden. DeepISP comprises two main stages: low-level stages and high-level stages. In the low-level stage, the focus lies on extracting low-level features from the input images and performing localized modifications [46]. Conversely, the high-level stage is tasked with extracting features at a higher level and executing global corrections [47]. Equation (5) is a quadratic function of the pixel's R, G, and B components. The triu (.) is the vectorized form of the elements, it is used to discard redundancies. This quadratic transformation produces pleasant-looking images.

$$W.\text{triu}\left([rgb1]^T.[rgb1]\right) \tag{5}$$

Equation (6) is a loss function used in DeepISP. The loss function is defined in the Lab domain. Since the network operates in the RGB color space, to calculate the loss, the network output needs to go through an RBG-to-Lab color conversion operator.

$$Loss(\hat{I}, I) = (1-\alpha)\|\text{Lab}(\hat{I}) - \text{Lab}(I)\|_1 + \alpha\, \text{MSSSIM}(L(\hat{I}), L(I)) \tag{6}$$

2.4. YOLOv8

The YOLO (You Only Look Once) method, developed by Ultralytics, is a widely utilized approach for object detection [48]–[50]. YOLOv8 is considered to be the new state-of-the-art in object detection, attributed to its higher mean Average Precision (mAP) scores and lower inference speed on the COCO dataset [51], [52]. The core of YOLO target detection lies in the compact model size and rapid computation speed [53], [54]. Its efficiency lies in the capability to perform prediction with just one forward propagation of the target image, significantly enhancing computational speed without compromising accuracy, making it well-suited for real-time applications [55], [56]. YOLO has undergone several iterations, with one of the latest versions being YOLOv8 [57], [58]. In comparison to its predecessors, YOLOv8 exhibits substantial performance improvements across various key tasks, such as object detection, semantic segmentation, and image classification [59], [60]. YOLOv8 stands out for its superior performance. The YOLO architecture consists of a backbone that converts image characteristics into feature maps of varying scales, a neck that manipulates these feature maps to produce consolidated and enhanced depictions, and a head that uses these refined attributes to generate predictions that include the bounding box and category labels [61]. The detailed architecture YOLOv8 is illustrated in Fig. 8.

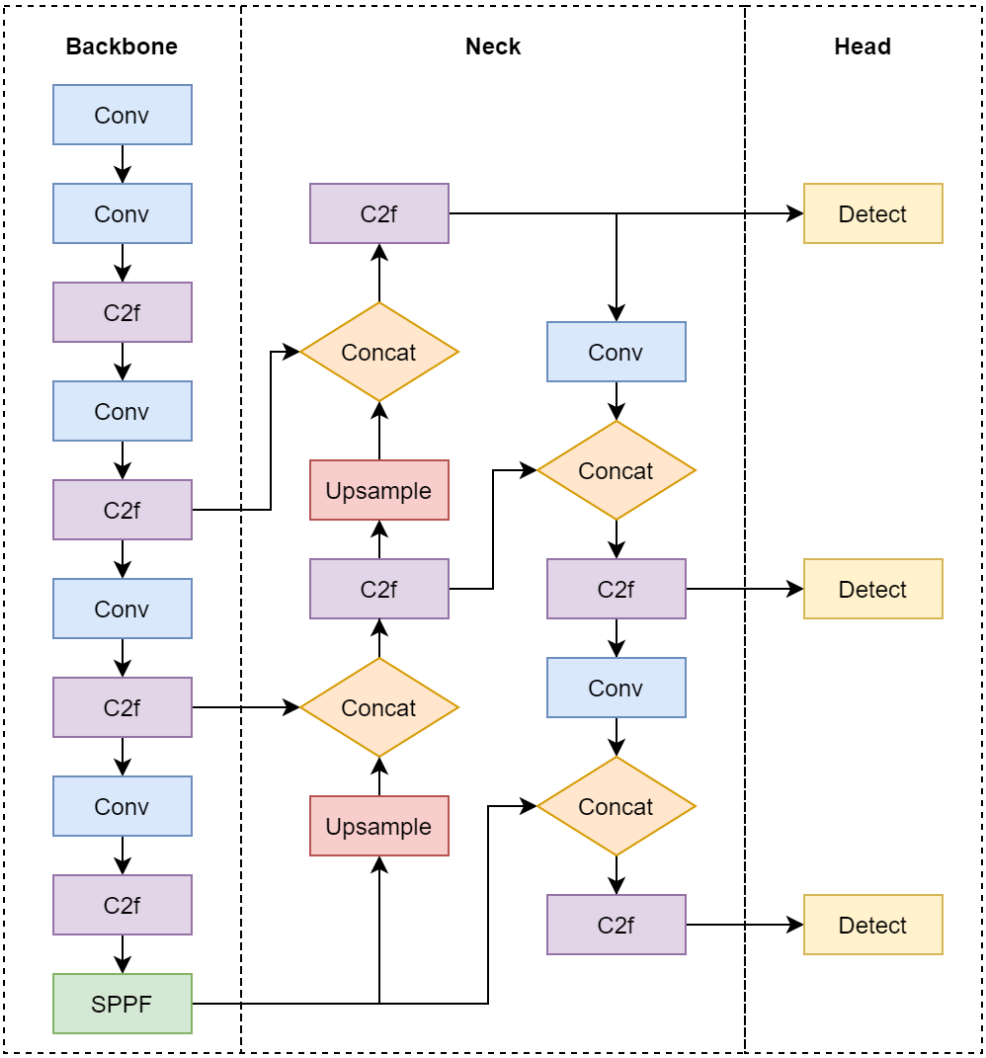


Fig. 8. YOLOv8 architecture

In this research, YOLOv8-N (You Only Look Once Version 8-N) was utilized as the core model for head movement detection. YOLOv8-N is part of the YOLO family of object detection models, known for its ability to detect and classify objects in real-time with high efficiency [62]. This version of YOLO is optimized for speed and performance, making it a suitable choice for real-time applications [63], such as controlling smart wheelchairs based on head pose estimation. As a state-of-the-art model, YOLOv8-N offers several advantages that make it well-suited for this application. Firstly, YOLOv8-N is renowned for its speed and efficiency, making it capable of processing images at high frame rates [64]. This is crucial for real-time applications such as head pose estimation in smart wheelchairs, where rapid detection is essential.

YOLOv8-N exhibits a high level of accuracy in detecting and classifying objects within images [65]. The model's ability to generate precise bounding boxes and accurate class predictions is vital for ensuring reliable head movement detection. This is particularly important in challenging environments with varying lighting conditions, where traditional object detection models may struggle. Additionally, YOLOv8-N maintains a balance between speed and accuracy, which is vital in real-time systems [52]. While its computational efficiency allows it to run on devices with limited resources, such as embedded systems in smart wheelchairs, it still provides accurate detection across a range of lighting conditions. In this study, YOLOv8-N's robustness was further enhanced by integrating image preprocessing techniques, which improved its accuracy in both normal-light and low-light environments, making it a versatile model for real-world assistive technology applications.

2.5. Training Process

The YOLOv8-N model was trained using pretrained weights for 100 epochs with an input image size of 128 $\times$ 128 pixels. Separate models were trained for each image preprocessing technique using the same training configuration to ensure a fair comparison. The batch size was set automatically by the YOLOv8 framework. Training checkpoints were saved at every epoch, and training performance plots were generated. All models were trained using the same configuration, differing only in the preprocessing technique applied to the input images. The training hyperparameters used in this study are summarized in Table 3.

Table 3. Training hyperparameters

Parameter	Value
Model variant	YOLOv8-N
Pretrained weights	Yes
Epochs	100
Input image size	128 $\times$ 128 pixels
Batch size	Auto
Optimizer	SGD
Initial learning rate	0.01
Rectangular training	True
Checkpoint saving	Every epoch
Training plots	Enabled

3. Results

3.1. Preprocessing Results

Test data samples captured under low-light conditions, after undergoing preprocessing, are illustrated in Fig. 9. This figure demonstrates the effectiveness of the preprocessing steps applied to enhance image quality. The images shown highlight the improvements made through resizing and enhancement techniques, such as Histogram Equalization, AutoContrast, and DeepISP, showcasing how these processes enhance visibility and detail in challenging lighting conditions.

Visual comparison of the resulting images after the preprocessing method was conducted to assess the quality of the image enhancement results produced by each preprocessing method. This evaluation aimed to ensure that the processed image had better clarity, contrast, and illumination than the original image, especially in low-light conditions. Through this testing, we assessed the extent to which each method was able to improve object visibility, emphasize important structures, and reduce noise that could potentially interfere with subsequent image analysis.

Fig. 9 illustrates the effectiveness of different image enhancement methods, comparing original images before preprocessing in Fig. 9 (a) and after applying the enhancement techniques in Fig. 9 (b), Fig. 9 (c), and Fig. 9 (d). Fig. 9 (b) demonstrates the results of the Histogram Equalization method, where the contrast and brightness are significantly improved. Fig. 9 (c) shows the images after applying AutoContrast method. Fig. 9 (d) presents images after preprocessing using the DeepISP method, highlighting enhanced brightness and clarity following the application. These figures collectively showcase the impact of each technique on image quality.

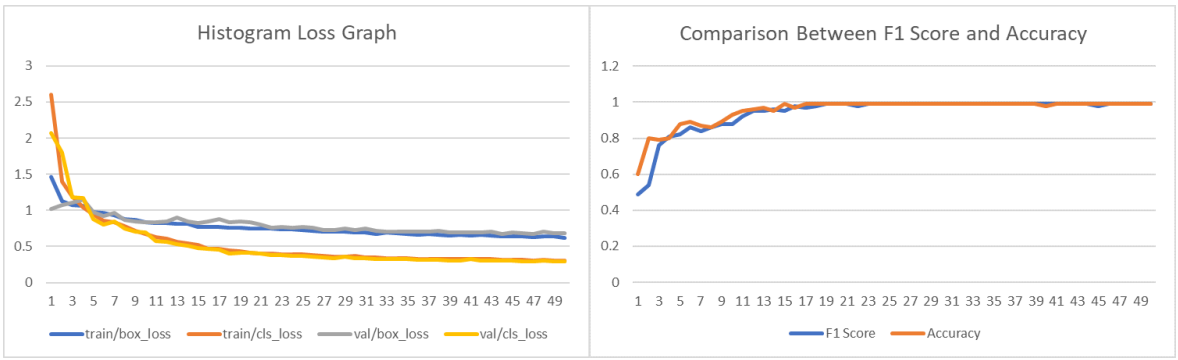
Fig. 9. Sample images after applying preprocessing methods: (a) original images (before preprocessing); (b) after Histogram Equalization; (c) after AutoContrast; (d) after DeepISP

3.2. Best Epoch

The optimal epoch for the YOLOv8-N model trained using the Histogram Equalization, AutoContrast, and DeepISP preprocessing methods, was determined based on the performance convergence point on the validation data, to achieve the best accuracy while preventing overfitting. The best epoch is identified by combining the loss and accuracy trends at each epoch. This determination was also made based on the lowest loss value, as well as the highest f1-score and accuracy. The outcomes of these tests are visually presented in Fig. 10 to Fig. 12, providing a comprehensive view of the model's performance across different epochs. In Fig. 10, the graphical representation illustrates the loss graph alongside the comparison graph of accuracy and f1-score for the model utilizing the Histogram Equalization method. Through the conducted tests, it was determined that epoch 16 emerges as the optimal epoch within the detection model employing the Histogram Equalization method. This epoch showcases a favorable balance between minimized loss and maximized accuracy and f1-score, indicating superior performance and efficacy of the model at this stage of training.

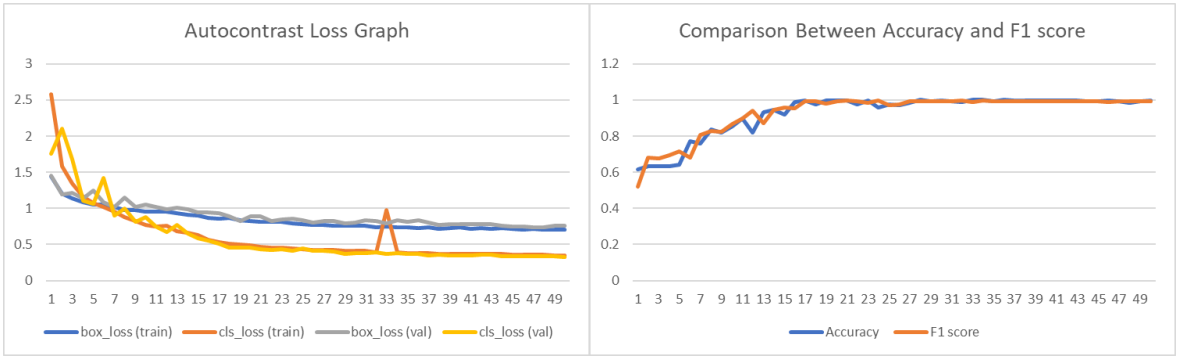
A graphical representation of the loss graph and the accuracy and f1-score comparison graph for the model using the AutoContrast enhancement method are shown in Fig. 11. Epoch 16 was found to be the best epoch in the detection model using the AutoContrast approach based on the

experiments that were carried out. The model's excellent performance and efficacy at this training stage are demonstrated by this epoch, which exhibits a favorable balance between reduced loss and maximal accuracy and f1-score. Fig. 12 illustrates the graphical representation of the loss graph alongside the comparison graph of accuracy and f1-score for the model utilizing the DeepISP method. Through the conducted tests, it was determined that epoch 17 emerges as the optimal epoch within the detection model employing the DeepISP method. This epoch showcases a favorable balance between minimized loss and maximized accuracy and f1-score, indicating superior performance and efficacy of the model at this stage of training.



(a) **(b)**

Fig. 10. Loss comparison graph (a) and accuracy comparison with f1-score (b) in the model using the Histogram Equalization preprocessing

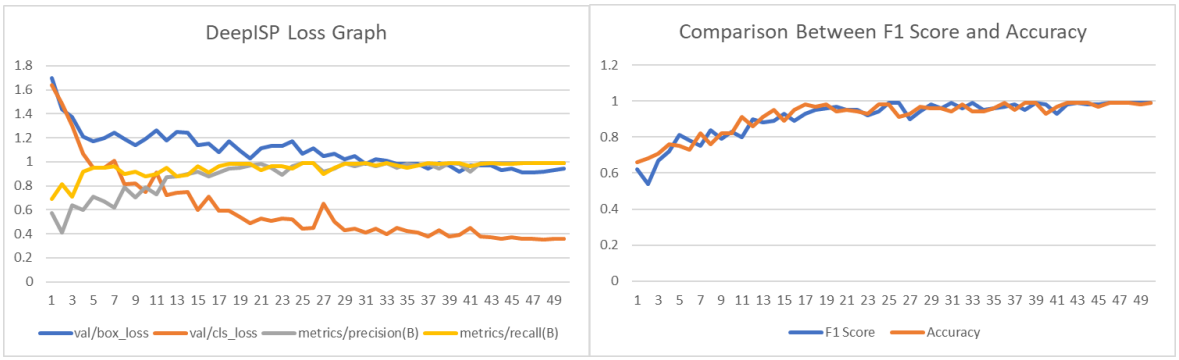


(a) **(b)**

Fig. 11. Loss comparison graph (a) and accuracy comparison with f1-score (b) in the model using the AutoContrast preprocessing

3.3. Performance Evaluation

We conducted performance evaluation to determine the enhancement method with the highest accuracy, while implemented in YOLOv8-N. Performance evaluations were conducted on four classes of head pose: front, down, right, and left, evaluated using evaluation metrics (accuracy, precision, recall, f1-score, and specificity). The purpose of this testing was to assess how each preprocessing technique affected detection stability, the model's capacity to differentiate between classes, and the system's resilience to changes in illumination. Thus, our evaluation aimed not only to improve accuracy but also to ensure the model's robustness in difficult lighting scenarios, notably in low-light environments.



(a) **(b)**

Fig. 12. Loss comparison graph (a) and accuracy comparison with f1-score (b) in the model using the DeepISP preprocessing

The results of the tests are presented in Table 4 to Table 7. Table 4 and Table 5 show performance results of the YOLOv8-N model without preprocessing under normal and low-light conditions. Meanwhile, Table 6 and Table 7 present a comparison of the performance results when applying the image preprocessing process using the Histogram Equalization, AutoContrast, and DeepISP methods.

Table 4. Performance per Class Without Preprocessing (Normal-Light)

Class	Accuracy	Precision	Recall	F1-Score	Specificity
Front	0.909	0.810	0.842	0.825	0.932
Down	0.942	0.891	0.891	0.891	0.961
Right	0.905	0.864	0.761	0.810	0.956
Left	0.863	0.621	0.774	0.689	0.885

Table 5. Performance per Class Without Preprocessing (Low-Light)

Class	Accuracy	Precision	Recall	F1-Score	Specificity
Front	0.875	0.786	0.750	0.767	0.922
Down	0.919	0.842	0.842	0.842	0.945
Right	0.831	0.667	0.650	0.658	0.892
Left	0.829	0.570	0.774	0.657	0.869

The evaluation metrics presented in Table 4 reveal performance per class of the YOLOv8-N model's under normal-light conditions. In general, the results obtained under normal lighting indicate that the model can accurately classify head pose across all classes, with the down class achieving the best accuracy (0.942) and specificity (0.961), demonstrating the model's capacity to identify movements with minimal mistake. The model's precision and sensitivity are also consistent, as shown by the comparatively balanced f1-score values, in the range of 0.81 and 0.89.

Table 5 shows performance per class under low-light condition without preprocessing method. It demonstrates that the model's performance declined across most metrics in low-light condition, especially in the right and left classes, where accuracy dropped to 0.831 and 0.829, respectively, while f1-scores only reached 0.658 and 0.657. This decline suggests that the YOLOv8-N model without preprocessing is still not as well suited to low-light situations as it could be. This is because visual features become less distinct, resulting in unreliable facial object and head pose identification.

In both lighting condition, the highest accuracy is in detecting downward head pose, while the lowest accuracy was observed when classifying head pose to the left. The model consistently achieved the highest accuracy in detecting downward head movements, indicating its robustness in this particular class.

Table 6. Performance per Class With Preprocessing (Normal-Light)

Method	Class	Accuracy	Precision	Recall	F1-Score	Specificity
Histogram	Front	0.915	0.857	0.826	0.841	0.948
Equalization	Down	0.957	0.916	0.916	0.916	0.971
	Right	0.880	0.769	0.746	0.758	0.925
	Left	0.854	0.622	0.807	0.703	0.885
AutoContrast	Front	0.928	0.857	0.865	0.861	0.949
	Down	0.964	0.931	0.931	0.931	0.976
	Right	0.915	0.872	0.798	0.833	0.957
	Left	0.883	0.674	0.807	0.734	0.901
DeepISP	Front	0.909	0.714	0.920	0.804	0.906
	Down	0.969	0.941	0.941	0.941	0.979
	Right	0.940	0.913	0.852	0.881	0.971
	Left	0.950	0.907	0.807	0.854	0.970

Table 7. Performance per Class With Preprocessing (Low-Light)

Method	Class	Accuracy	Precision	Recall	F1-Score	Specificity
Histogram	Front	0.918	0.848	0.840	0.844	0.946
Equalization	Down	0.964	0.931	0.931	0.931	0.976
	Right	0.880	0.749	0.756	0.753	0.919
	Left	0.900	0.725	0.807	0.764	0.916
AutoContrast	Front	0.885	0.705	0.831	0.763	0.900
	Down	0.956	0.916	0.916	0.916	0.971
	Right	0.913	0.872	0.791	0.829	0.957
	Left	0.903	0.777	0.807	0.792	0.930
DeepISP	Front	0.908	0.738	0.891	0.807	0.912
	Down	0.944	0.891	0.891	0.891	0.962
	Right	0.943	0.923	0.853	0.887	0.975
	Left	0.938	0.881	0.807	0.843	0.962

On the other hand, the detection of head turns to the left presented the greatest challenge for the model, as it consistently showed the lowest accuracy in both normal and low lighting conditions. This lower accuracy could be due to the similarity in visual cues between head turns to the left and other lateral movements, making it more difficult for model to differentiate between them. This finding suggests that further optimization may be necessary to improve the model's performance in recognizing leftward head movements. Overall, the results from Table III and IV show that different lighting conditions have a substantial effect on model performance, especially in classes with large variations in lighting and head pose. This lays the groundwork for implementing illumination-enhancing preprocessing methods in the next stage, which will improve the model's robustness to varying lighting circumstances.

The application of different preprocessing methods (Histogram Equalization, AutoContrast, and DeepISP) were evaluated for their impact on the YOLOv8-N model's performance in head pose estimation. The evaluation metrics results are presented in Table 6 and Table 7, reveal significant variations in the YOLOv8-N model's performance in classifying head pose when different preprocessing techniques are applied. Table 6 presents performance in normal-light condition, while Table 7 shows performance in low-light condition.

In normal-light condition, as presented in Table 6, all three preprocessing methods demonstrated improved performance compared to the model without preprocessing, particularly in accuracy and f1-score, indicate better detection stability. Histogram Equalization and AutoContrast consistently achieved higher precision and f1-score values in most classes, particularly the down class, where both methods obtained accuracy greater than 0.95. The Histogram Equalization method produces consistent results across classes, with the down class outperforming the others (accuracy of 0.957).

This shows its capacity to balance illumination levels while maintaining crucial details in the image. AutoContrast also produces competitive results, including the highest performance in the down class (accuracy of 0.964), indicating its ability to improve contrast.

Meanwhile, DeepISP preprocessing resulted in the highest performance in most classes across all metrics. Despite showing variations among classes, it still has the highest recall value (0.941 in the down class), proving its superiority in keeping significant information in locations with challenging lighting condition. In comparison to the data obtained without preprocessing, recall increases by an average of 7-10%, showing enhanced pose detection sensitivity. DeepISP achieved competitive results with strong performance, with accuracy greater than 0.9 in all classes, demonstrating its capacity to preserve critical visual information during the lighting enhancement process. This indicates that DeepISP significantly enhances the model's ability to correctly identify head movements under optimal lighting conditions.

A similar trend is observed under low-light conditions, as shown in Table 7. The benefits of applying preprocessing were even more significant in low-light condition. The implementation of AutoContrast and DeepISP significantly increased detection stability over the model without preprocessing. Histogram Equalization also contributed to a more balanced performance gain across all classes, although its precision is slightly lower than the other two methods. Once again, DeepISP preprocessing led to the highest accuracy and overall performance when compared to Histogram Equalization and AutoContrast. DeepISP performance was especially noticeable in the right and left classes. In the right class, the accuracy reached 0.943 with an f1-score of 0.887, whereas in the left class, the accuracy and f1-score were 0.938 and 0.843, respectively. This demonstrates that DeepISP is more adaptable to lighting variations, particularly when light intensity varies.

The comparative analysis of Table 6 and Table 7 reveals that DeepISP consistently outperforms Histogram Equalization and AutoContrast in enhancing the YOLOv8-N model's performance in most classes across both normal-light and low-light conditions. The significant improvements observed in accuracy, precision, recall, f1-score, and specificity demonstrate the crucial role of DeepISP in extracting informative features for accurate head movement classification. This suggests that DeepISP is particularly well-suited for applications requiring reliable head pose estimation in environments with varying lighting conditions.

Table 8 and Table 9 shows average performance comparison across enhancement methods in normal-light and low-light conditions. Under normal lighting conditions, as shown in Table 8, the DeepISP method outperformed all other methods and the condition without preprocessing, with the highest value of all evaluation metrics: accuracy of 0.942, precision of 0.869, recall of 0.880, f1-score of 0.870, and specificity of 0.957. This demonstrates that DeepISP is effective at maintaining image detail while adaptively enhancing illumination quality. AutoContrast likewise produced competitive results, with a solid balance of precision (0.834) and recall (0.850), although Histogram Equalization was slightly lower but still gave a consistent improvement over no preprocessing.

Table 8. Average performance comparison across enhancement methods in normal-light condition

Method	Accuracy	Precision	Recall	F1-Score	Specificity
Without preprocessing	0.905	0.797	0.817	0.804	0.934
Histogram Equalization	0.902	0.791	0.824	0.805	0.932
AutoContrast	0.923	0.834	0.850	0.840	0.946
DeepISP	**0.942**	**0.869**	**0.880**	**0.870**	**0.957**

Under low lighting conditions, as in Table 9, DeepISP again achieved the highest evaluation metrics, with an accuracy of 0.933, precision of 0.858, recall of 0.861, f1-score of 0.857, and specificity of 0.953, demonstrating its superior capacity to withstand low image quality caused by low-light environments.

Table 9. Average performance comparison across enhancement methods in low-light condition

Method	Accuracy	Precision	Recall	F1-Score	Specificity
Without preprocessing	0.864	0.716	0.754	0.731	0.907
Histogram Equalization	0.916	0.813	0.834	0.823	0.939
AutoContrast	0.914	0.818	0.836	0.825	0.940
DeepISP	**0.933**	**0.858**	**0.861**	**0.857**	**0.953**

Meanwhile, Histogram Equalization and AutoContrast produced moderate improvements over no preprocessing. Overall, these findings show that DeepISP is the most effective strategy for boosting model performance across both illumination conditions. These findings highlight the importance of lighting correction in improving model robustness and reliability in real-world smart wheelchair navigation scenarios.

In our previous work [21], using the same dataset, an accuracy of 0.873 was achieved under laboratory conditions through offline evaluation. In another study [24], an accuracy of 0.93 was reported using a different approach. However, the evaluation was also conducted exclusively in a laboratory setting. Compared to these approaches, the proposed system achieves comparable accuracy while additionally supporting real-time operation and on-device deployment on a smart wheelchair platform, demonstrating improved practical applicability in real-world scenarios.

This research demonstrates the significant potential of image preprocessing techniques, particularly DeepISP, to enhance the performance of head-controlled wheelchair systems, especially in low-light environments. By addressing the challenges caused by poor lighting, DeepISP enables more accurate and reliable head pose estimation. This is crucial for ensuring the safety and efficiency of real-time applications such as smart wheelchairs.

3.4. Computation Time

Testing was also conducted to identify the method with the fastest computing speed. It is crucial to understand the trade-off between performance and computation time, considering that the system is used for real-time head pose estimation as smart wheelchair navigation. All three models were executed under identical conditions and on the same device, to ensure a fair comparison of performance, eliminating potential variations caused by hardware differences or environmental factors.

The results of the computing time speed for the three preprocessing methods are presented in Table 10. The analysis of the data presented in Table 10 reveals that the Histogram Equalization method yields the fastest average computational speed (27.78 ms) when integrated with the YOLOv8-N model for head pose estimation, outperforming other enhancement methods.

This method significantly enhances the responsiveness of the system, making it suitable for applications requiring timely head movement recognition. On the other hand, the longest processing time for head pose estimation occurs when DeepISP is used in the preprocessing stage, taking an average of 34.82 ms. Although slightly slower, this trade-off comes with notable improvements in accuracy and robustness, making DeepISP an effective option despite its higher processing time.

Moreover, the computational cost associated with DeepISP, while slightly higher than AutoContrast and Histogram Equalization, remains within an acceptable range for real-time applications. The average processing time of 34.82 ms is sufficiently fast for integration into a smart wheelchair system. This suggests that the benefits gained in terms of improved accuracy outweigh the marginal increase in computational overhead.

4. Discussion

The experimental results demonstrate that image illumination enhancement plays a critical role in improving the performance of vision-based head-controlled wheelchair systems.

Table 10. Comparison of computation time

Test	Histogram Equalization (ms)	AutoContrast (ms)	DeepISP (ms)
1	27.5	34.2	35.9
2	26.8	32.5	36.0
3	27.0	35.4	30.2
4	27.7	30.9	34.9
5	29.9	36.1	36.1
6	27.6	28.4	35.6
7	27.5	30.4	35.3
8	28.3	28.2	34.6
Average	**27.8**	**32.01**	**34.82**

Among the evaluated preprocessing techniques, DeepISP consistently outperformed Autocontrast and Histogram Equalization across both normal-light and low-light conditions, achieving the highest accuracy while maintaining acceptable computational cost. Although DeepISP introduces slightly higher processing time, the observed latency remains within real-time constraints, making it suitable for practical wheelchair deployment. These findings confirm that the accuracy gains provided by DeepISP outweigh the marginal increase in computational overhead, particularly in challenging lighting environments.

4.1. Main Findings

The implementation of DeepISP as a preprocessing technique has shown significant advantages in enhancing the performance of the YOLOv8-N model for head pose estimation. By effectively addressing varying lighting conditions, DeepISP extracts more informative visual features, enabling the model to produce more accurate and reliable classifications. This improvement is clearly reflected in the higher accuracy achieved in both normal-light and low-light environments. Overall, the results confirm that image illumination enhancement plays a critical role in improving the performance of vision-based head-controlled wheelchair systems.

4.2. Comparison with Other Studies

Compared to previous vision-based smart wheelchair systems that primarily focused on head pose detection under controlled lighting conditions [17], [18], the proposed method demonstrates superior robustness in low-light environments. While earlier studies reported promising accuracy under normal lighting, their performance was not evaluated under varying illumination conditions.

In our previous work [21], using the same dataset, an accuracy of 0.873 was achieved under laboratory conditions through offline evaluation. In another study [24], an accuracy of 0.93 was reported using a different approach; however, the evaluation was also conducted exclusively in a laboratory setting. Compared to these approaches, the proposed system achieves comparable accuracy while additionally supporting real-time operation and on-device deployment on a smart wheelchair platform. This highlights a key advancement over previous studies by extending performance evaluation from laboratory settings to real-world implementation.

4.3. Implications

One of the key advantages of DeepISP lies in its ability to enhance image quality by optimizing contrast, exposure, and noise reduction, which leads to improved feature extraction during the detection process. This capability is particularly beneficial in challenging lighting conditions, such as low-light environments, where conventional image enhancement methods often struggle to preserve critical visual details. By mitigating the effects of poor lighting, DeepISP ensures that the YOLOv8-N model can consistently produce accurate head pose classifications even in suboptimal visual environments. From an application perspective, these findings imply that learning-based illumination enhancement

is essential for deploying vision-based assistive systems in real-world scenarios, where lighting conditions are unpredictable. The improved robustness directly enhances the usability, reliability, and safety of head-controlled smart wheelchair systems.

From a practical perspective, the improved accuracy and robustness achieved through illumination enhancement directly benefit real-world assistive technology applications. By maintaining stable head pose detection under varying lighting conditions, the proposed system reduces the risk of misclassification that could lead to unintended wheelchair movements. This increased reliability is particularly important for daily use in uncontrolled environments, such as indoor spaces with insufficient lighting or outdoor settings with uneven illumination, thereby enhancing user safety, confidence, and overall usability of head-controlled smart wheelchairs.

4.4. Strength and Limitations

A major strength of this study is the real-time implementation of the proposed system on an actual smart wheelchair platform, rather than relying solely on offline laboratory evaluation. In addition, the systematic comparison of multiple illumination enhancement techniques provides a clear understanding of the trade-offs between accuracy and computational cost. Although DeepISP introduces slightly higher computational overhead than Histogram Equalization and AutoContrast, the average processing time of 34.82 ms remains within acceptable limits for real-time operation on embedded platforms. This indicates that the accuracy improvements achieved by DeepISP outweigh the marginal increase in computational cost.

Nevertheless, this study has certain limitations. The dataset was collected under a limited range of environments, which may not fully represent all real-world lighting variations. Furthermore, while DeepISP demonstrated superior performance, its higher computational demand may pose challenges for deployment on lower-end hardware. Future work will focus on expanding dataset diversity and exploring more lightweight enhancement models to further optimize system efficiency. This trade-off indicates that although DeepISP is well-suited for real-time smart wheelchair systems, further optimization may be required for scenarios with stricter latency or hardware resource constraints.

5. Conclusion

This study focuses on comparing three image enhancement methods—Histogram Equalization, AutoContrast, and DeepISP, as preprocessing steps to improve head pose estimation performance in a YOLOv8-N-based smart wheelchair system. To ensure performance in a real-world operational context, the evaluation included not only accuracy but also computational time using real hardware integrated into a smart wheelchair. Experimental results showed that implementing DeepISP as a preprocessing step provides the most significant accuracy improvement under both lighting conditions (normal-light and low-light), achieving accuracies of 0.942 and 0.933, respectively, surpassing AutoContrast and Histogram Equalization, as well as the scenario without preprocessing.

Although DeepISP required slightly longer computational time (34.82 ms) compared to AutoContrast (32.01 ms) and Histogram Equalization (27.78 ms), the observed latency remains within acceptable limits for real-time wheelchair control. This result highlights a trade-off between detection accuracy and processing speed, where the accuracy gain outweighs the marginal increase in computational overhead and contributes to safer and more reliable wheelchair operation by reducing the risk of unintended movements under challenging lighting conditions. One of the key advantages of DeepISP lies in its ability to enhance image quality by optimizing contrast, exposure, and noise reduction, allowing YOLOv8-N to extract important features more reliably. These findings indicate that learning-based illumination enhancement is more effective than traditional methods for head pose estimation under varying lighting conditions and contributes new insights into illumination-robust perception for vision-based assistive systems.

Nevertheless, this study has several limitations, including a limited dataset size, constrained environmental diversity, and dependency on predefined lighting thresholds. Future work will focus on optimizing computational efficiency for deployment on lower-end hardware, expanding evaluations to more diverse users and environments, and validating the proposed approach using public benchmark datasets such as BIWI under varied illumination conditions. Beyond smart wheelchairs, the proposed framework may also be applicable to other vision-based robotic and assistive systems that require robust perception in low-light environments.

Author Contribution: All authors contributed equally to the main contributor to this paper. All authors read and approved the final paper.

Funding: This research was funded by the Ministry of Education and Culture, Research and Technology, Directorate General of Higher Education Research and Technology, Republic of Indonesia. Research grant number 00309.93/UN10.A0501/B/PT.O1.03.2/2024.

Acknowledgment: We would like to express our gratitude to the Ministry of Education and Culture, Research and Technology, Directorate General of Higher Education Research and Technology, Republic of Indonesia (grant number 00309.93/UN10.A0501/B/PT.O1.03.2/2024) for supporting this research.

Conflicts of Interest: The authors declare no conflict of interest.

References

[1] WHO, *Health equity for persons with disabilities*, World Health Organization, 2024, https://books.google.co.id/books?id=OVswEQAAQBAJ&hl=id&source=gbs_navlinks_s.

[2] S. Pancholi, J. P. Wachs, and B. S. Duerstock, "Use of artificial intelligence techniques to assist individuals with physical disabilities," *Annual Review of Biomedical Engineering*, vol. 26, pp. 1–24, 2024, https://doi.org/10.1146/annurev-bioeng-082222.

[3] W. Al Syifa and E. N. Hadi, "Determinants of quality of life on persons with physical disability: literature review," *Journal of Social Research*, vol. 2, pp. 1786–1795, 2023, https://doi.org/10.55324/josr.v2i6.914.

[4] G. McKenzie, C. Willis, and N. Shields, "Barriers and facilitators of physical activity participation for young people and adults with childhood-onset physical disability: a mixed methods systematic review," *Dev Med Child Neurol*, vol. 65, no. 3, pp. 914–925, 2021, https://doi.org/10.1111/dmcn.15464.

[5] M. Asirdizer, Y. Hekimoğlu, and S. Keskin, "Investigation of effective factors on traumatic amputations due to road traffic accidents," *Injury*, vol. 53, no. 3, pp. 966–971, 2022, https://doi.org/10.1016/j.injury.2021.11.021.

[6] N. Jayasekera *et al*., "Revolutionizing accessibility: smart wheelchair robot and mobile application for mobility, assistance, and home management," *Journal of Robotics and Control (JRC)*, vol. 5, no. 1, pp. 27–53, 2024, https://doi.org/10.18196/jrc.v5i1.20057.

[7] F. B. Haque, T. H. Shuvo and R. Khan, "Head Motion Controlled Wheelchair for Physically Disabled People," *2021 Second International Conference on Smart Technologies in Computing, Electrical and Electronics (ICSTCEE)*, pp. 1-6, 2021, https://doi.org/10.1109/ICSTCEE54422.2021.9708577.

[8] W. Hutamaputra, F. Utaminingrum, A. S. Budi, and K. Ogata, "Eyes gaze detection based on multiprocess of ratio parameters for smart wheelchair menu selection in different screen size," *Journal of Visual Communication and Image Representation*, vol. 91, pp. 1–8, 2023, https://doi.org/10.1016/j.jvcir.2023.103756.

[9] F. Utaminingrum, I. K. Somawirata, G. Pengestu, T. Thaipisutikul, and T. K. Shih, "Selecting control menu on electric wheelchair using eyeball movement for diffable person," *International Journal on Informatics Visualization*, vol. 7, no. 1, pp. 37–43, 2023, https://doi.org/10.30630/joiv.7.1.1011.

[10] A. M. Alqadri and F. Utaminingrum, "Lightweight YOLOv5-Based Algorithm to Detect Room Nameplates for Autonomous Smart Wheelchair," *2024 21st International Joint Conference on Computer Science and Software Engineering (JCSSE)*, pp. 463-468, 2024, https://doi.org/10.1109/JCSSE61278.2024.10613722.

[11] M. Joly, A. Pradeep, and K. S, "Intelligent voice controlled wheel chair for disabled people," *Cardiometry*, no. 26, pp. 532–536, 2023, https://doi.org/10.18137/cardiometry.2023.26.532536.

[12] A. I. Iskanderani *et al*., "Voice Controlled Artificial Intelligent Smart Wheelchair," *2020 8th International Conference on Intelligent and Advanced Systems (ICIAS)*, pp. 1-5, 2021, https://doi.org/10.1109/ICIAS49414.2021.9642607.

[13] M. Nishimori, T. Saitoh and R. Konishi, "Voice Controlled Intelligent Wheelchair," *SICE Annual Conference*, pp. 336-340, 2007, https://doi.org/10.1109/SICE.2007.4421003.

[14] S. Hariharan, A. Abinaya, V. Anjuga, and V. Bhuvaneshwari, "Voice controlled wheelchair for physically disabled people and blind people," *Assian Journal of Applied Science and Technology*, vol. 9, no. 1, pp. 21–28, 2025, https://doi.org/10.38177/ajast.2025.9103.

[15] J. Sjah *et al*., "Digital image correlation for the determination of mechanical properties of concrete with modified expanded polystyrene," *International Journal of Technology*, vol. 15, no. 2, p. 394, 2024, https://dx.doi.org/10.14716/ijtech.v15i2.6697.

[16] M. A. B. Sarker, E. Sola-Thomas, C. Jamieson, and M. H. Imtiaz, "Autonomous movement of wheelchair by cameras and YOLOv7," *Engineering Proceedings*, vol. 31, no. 1, p. 60, 2023, https://doi.org/10.3390/ASEC2022-13834.

[17] I. K. Somawirata and F. Utaminingrum, "Smart wheelchair controlled by head gesture based on vision," in *Journal of Physics: Conference Series*, vol. 2497, 2023, https://doi.org/10.1088/1742-6596/2497/1/012011.

[18] S. Chatzidimitriadis, S. M. Bafti and K. Sirlantzis, "Non-Intrusive Head Movement Control for Powered Wheelchairs: A Vision-Based Approach," in *IEEE Access*, vol. 11, pp. 65663-65674, 2023, https://doi.org/10.1109/ACCESS.2023.3275529.

[19] X. Li, Z. Li, L. Zhou, and Z. Huang, "FOLD: low-level image enhancement for low-light object detection based on FPGA MPSoC," *Electronics*, vol. 13, no. 1, p. 230, 2024, https://doi.org/10.3390/electronics13010230.

[20] J. Wang *et al*., "Research on improved YOLOv5 for low-light environment object detection," *Electronics*, vol. 12, no. 14, p. 3089, 2023, https://doi.org/10.3390/electronics12143089.

[21] A. F. Rahmaniati and F. Utaminingrum, "Deep Learning Based Smart Wheelchair Navigation Optimization for Multi-Lighting Conditions," *2024 4th International Conference on Robotics, Automation and Artificial Intelligence (RAAI)*, pp. 295-300, 2024, https://doi.org/10.1109/RAAI64504.2024.10949523.

[22] A. Tonge and S. Thepade, "ViSTORY: effective video storyboard generation with visual keyframes using discrete cosine transform," *International Journal of Technology*, vol. 14, no. 2, pp. 411–421, 2023, https://doi.org/10.14716/ijtech.v14i2.5617.

[23] S. U. A. Shovo, M. G. R. Abir, M. M. Kabir, and M. F. Mridha, "Advancing low-light object detection with you only look once models: an empirical study and performance evaluation," *Cognitive Computation and Systems*, vol. 6, no. 4, pp. 119–134, 2024, https://doi.org/10.1049/ccs2.12114.

[24] A. F. Rahmaniati, F. Utaminingrum, and B. D. Setiawan, "Optimizing head movement classification under varying lighting using spatial and channel attention in EfficientNetV2," in *Lecture Notes on Data Engineering and Communications Technologies*, vol. 260, pp. 47–58, 2025, https://doi.org/10.1007/978-3-031-96099-4_5.

[25] Y. Afaq and A. Manocha, "Blockchain and deep learning integration for various application: a review," *Journal of Computer Information Systems*, vol. 64, no. 1, pp. 92–105, 2024, https://doi.org/10.1080/08874417.2023.2173330.

[26] M. Hussain, "YOLOv1 to v8: Unveiling Each Variant–A Comprehensive Review of YOLO," in *IEEE Access*, vol. 12, pp. 42816-42833, 2024, https://doi.org/10.1109/ACCESS.2024.3378568.

[27] J. Terven, D. M. Córdova-Esparza, and J. A. Romero-González, "A comprehensive review of YOLO architectures in computer vision: from YOLOv1 to YOLOv8 and YOLO-NAS," *Machine Learning and Knowledge Extraction*, vol. 5, no. 4, pp. 1680–1716, 2023, https://doi.org/10.3390/make5040083.

[28] R. F. A. Bonifacio and C. D. R. Peña, "Robotics and Computer Vision in Precision Agriculture: A Systematic Review of Applications and Technology Readiness," *International Journal of Robotics and Control Systems*, vol. 5, no. 4, pp. 2246–2264, 2025, https://doi.org/10.31763/ijrcs.v5i4.2218.

[29] V.-H. Le *et al*., "Improved multi-layer fusion framework based on loss function for visual odometry estimation from color image sequence," *International Journal of Technology*, vol. 16, no. 4, pp. 1265–1282, 2025, https://doi.org/10.14716/ijtech.v16i4.7517.

[30] J. Giner-Miguelez, A. Gómez, and J. Cabot, "A domain-specific language for describing machine learning datasets," *Journal of Computer Languages*, vol. 76, p. 101209, 2023, https://doi.org/10.1016/j.cola.2023.101209.

[31] C. Fan, M. Chen, X. Wang, J. Wang, and B. Huang, "A review on data preprocessing techniques toward efficient and reliable knowledge discovery from building operational data," *Frontiers in Energy Research*, vol. 9, 2021, https://doi.org/10.3389/fenrg.2021.652801.

[32] S. Roy, K. Bhalla, and R. Patel, "Mathematical analysis of histogram equalization techniques for medical image enhancement: a tutorial from the perspective of data loss," *Multimedia Tools and Applications*, vol. 83, no. 5, pp. 14363–14392, 2024, https://doi.org/10.1007/s11042-023-15799-8.

[33] P. Härtinger and C. Steger, "Adaptive histogram equalization in constant time," *Journal of Real-Time Image Proces*, vol. 21, no. 93, 2024, https://doi.org/10.1007/s11554-024-01465-1.

[34] S. Saifullah and R. Drezewski, "Modified histogram equalization for improved CNN medical image segmentation," in *Procedia Computer Science*, vol. 225, pp. 3021–3030, 2023, https://doi.org/10.1016/j.procs.2023.10.295.

[35] M. Mehdizadeh, K. T. Tafti, and P. Soltani, "Evaluation of histogram equalization and contrast limited adaptive histogram equalization effect on image quality and fractal dimensions of digital periapical radiographs," *Oral Radiology*, vol. 39, pp. 418–424, 2023, https://doi.org/10.1007/s11282-022-00654-7.

[36] B. D. Satoto, M. I. Utoyo, R. Rulaningtyas and E. B. Koendhori, "An auto contrast custom convolutional neural network to identifying Gram-negative bacteria," *2020 International Conference on Computer Engineering, Network, and Intelligent Multimedia (CENIM)*, pp. 70-75, 2020, https://doi.org/10.1109/CENIM51130.2020.9297964.

[37] J. Hou, G. Gendy, and G. He, "Phydiisp: a physics-guided differentiable pipeline for low-light machine vision," *Signal Image Video Process*, vol. 19, no. 35, 2025, https://doi.org/10.1007/s11760-025-03918-x.

[38] K. -H. Uhm, K. Choi, S. -W. Jung and S. -J. Ko, "Image Compression-Aware Deep Camera ISP Network," in *IEEE Access*, vol. 9, pp. 137824-137832, 2021, https://doi.org/10.1109/ACCESS.2021.3116702.

[39] Q. Yuan and S. Dai, "Adaptive histogram equalization with visual perception consistency," *Information Sciences*, vol. 668, p. 120525, 2024, https://doi.org/10.1016/j.ins.2024.120525.

[40] R. M. Dyke and K. Hormann, "Histogram equalization using a selective filter," *Visual Computer*, vol. 39, pp. 6221–6235, 2023, https://doi.org/10.1007/s00371-022-02723-8.

[41] J. Wang, H. Wang, Y. Sun, and J. Yang, "Improved Retinex-Theory-Based Low-Light Image Enhancement Algorithm," *Applied Sciences*, vol. 13, no. 14, p. 8148, 2023, https://doi.org/10.3390/app13148148.

[42] K. G. Dhal, A. Das, S. Ray, J. Gálvez, and S. Das, "Histogram equalization variants as optimization problems: a review," *Archives of Computational Methods in Engineering*, vol. 28, pp. 1471–1496, 2021, https://doi.org/10.1007/s11831-020-09425-1.

[43] R. -C. Chen, C. Dewi, Y. -C. Zhuang and J. -K. Chen, "Contrast Limited Adaptive Histogram Equalization for Recognizing Road Marking at Night Based on Yolo Models," in *IEEE Access*, vol. 11, pp. 92926-92942, 2023, https://doi.org/10.1109/ACCESS.2023.3309410.

[44] I. -J. Yu, S. -H. Nam, W. Ahn, M. -J. Kwon and H. -K. Lee, "Manipulation Classification for JPEG Images Using Multi-Domain Features," in *IEEE Access*, vol. 8, pp. 210837-210854, 2020, https://doi.org/10.1109/ACCESS.2020.3037735.

[45] J. Li *et al*., "Image-Scaling Attack on Image Signal Processing Pipelines in Deep Neural Networks-Based Outdoor Vision Applications," in *IEEE Transactions on Consumer Electronics*, vol. 70, no. 4, pp. 7044-7055, 2024, https://doi.org/10.1109/TCE.2024.3413720.

[46] C. F. G. dos Santos *et al*., "ISP meets deep learning: a survey on deep learning methods for image signal processing," *ACM Computing Surveys*, vol. 57, no. 5, pp. 1–44, 2025, https://doi.org/10.1145/3708516.

[47] E. Schwartz, R. Giryes and A. M. Bronstein, "DeepISP: Toward Learning an End-to-End Image Processing Pipeline," in *IEEE Transactions on Image Processing*, vol. 28, no. 2, pp. 912-923, 2019, https://doi.org/10.1109/TIP.2018.2872858.

[48] P. Jiang, D. Ergu, F. Liu, Y. Cai, and B. Ma, "A review of YOLO algorithm developments," *Procedia Computer Science*, vol. 199, pp. 1066–1073, 2022, https://doi.org/10.1016/j.procs.2022.01.135.

[49] K. Yamtuan, T. Radomngam, and P. Prempraneerach, "Visual servo kinematic control of delta robot using YOLOv5 algorithm," *Journal of Robotics and Control (JRC)*, vol. 4, no. 6, pp. 818–831, 2023, https://doi.org/10.18196/jrc.v4i6.19102.

[50] A. N. Handayani *et al*., "Real-Time Obstacle Detection for Unmanned Surface Vehicle Maneuver," *International Journal of Robotics and Control Systems*, vol. 3, no. 4, pp. 765–779, 2023, https://doi.org/10.31763/ijrcs.v3i4.1147.

[51] D. Reis, J. Kupec, J. Hong, and A. Daoudi, "Real-time flying object detection with YOLOv8," *arXiv*, 2024, https://doi.org/10.48550/arXiv.2305.09972.

[52] R. Varghese and S. M, "YOLOv8: A Novel Object Detection Algorithm with Enhanced Performance and Robustness," *2024 International Conference on Advances in Data Engineering and Intelligent Computing Systems (ADICS)*, pp. 1-6, 2024, https://doi.org/10.1109/ADICS58448.2024.10533619.

[53] M. Safaldin, N. Zaghden and M. Mejdoub, "An Improved YOLOv8 to Detect Moving Objects," in *IEEE Access*, vol. 12, pp. 59782-59806, 2024, https://doi.org/10.1109/ACCESS.2024.3393835.

[54] B. Xiao, M. Nguyen, and W. Q. Yan, "Fruit ripeness identification using YOLOv8 model," *Multimedia Tools and Applications*, vol. 83, pp. 28039–28056, 2024, https://doi.org/10.1007/s11042-023-16570-9.

[55] F. Huang, S. Chen, Q. Wang, Y. Chen, and D. Zhang, "Using deep learning in an embedded system for real-time target detection based on images from an unmanned aerial vehicle: vehicle detection as a case study," *International Journal of Digital Earth*, vol. 16, no. 1, pp. 910–936, 2023, https://doi.org/10.1080/17538947.2023.2187465.

[56] N. Rohaziat, M. R. M. Tomari, W. N. W. Zakaria, and D. Das, "Lightweight white blood cells detection using fusion of YOLOv5 and attention model," *Journal of Advanced Research in Applied Sciences and Engineering Technology*, vol. 48, no. 1, pp. 117–136, 2025, https://doi.org/10.37934/araset.48.1.117136.

[57] C.-Y. Wang and H.-Y. M. Liao, "YOLOv1 to YOLOv10: the fastest and most accurate real-time object detection systems," *APSIPA Transactions on Signal and Information Processing*, vol. 13, no. 1, 2024, https://doi.org/10.1561/116.20240058.

[58] F. M. Talaat and H. ZainEldin, "An improved fire detection approach based on YOLO-v8 for smart cities," *Neural Computing and Applications*, vol. 35, pp. 20939–20954, 2023, https://doi.org/10.1007/s00521-023-08809-1.

[59] L. Zhang, B. Li, Y. Cui, Y. Lai, and J. Gao, "Research on improved YOLOv8 algorithm for insulator defect detection," *Journal of Real-Time Image Processing*, vol. 21, no. 22, 2024, https://doi.org/10.1007/s11554-023-01401-9.

[60] M. Elavarasu and K. Govindaraju, "Unveiling the advancements: YOLOv7 vs YOLOv8 in pulmonary carcinoma detection," *Journal of Robotics and Control (JRC)*, vol. 5, no. 2, pp. 459–470, 2024, https://doi.org/10.18196/jrc.v5i2.20900.

[61] A. Baharuddin and M. A. M. Basri, "A YOLO-Based Target Detection Algorithm for DJI Tello Drone," *International Journal of Robotics and Control Systems*, vol. 5, no. 3, pp. 1608–1624, 2025, https://doi.org/10.31763/ijrcs.v5i3.1898.

[62] S. Ling *et al*., "Accurate recognition of jujube tree trunks based on CLAHE image enhancement and improved YOLOv8," *Research Square*, vol. 1, 2023, https://doi.org/10.21203/rs.3.rs-3240060/v1.

[63] Y. Shao, D. Zhang, H. Chu, X. Zhang, and Y. Rao, "A review of YOLO object detection based on deep learning," *Journal of Electronic Information Technology*, vol. 44, no. 10, pp. 3697–37008, 2022, https://doi.org/10.11999/JEIT210790.

[64] A. M. Alqadri *et al*., "Attention Module in YOLO-Based Object Detection Method for Autonomous Smart Wheelchair Room Navigation System," *2024 4th International Conference on Robotics, Automation and Artificial Intelligence (RAAI)*, pp. 312-316, 2024, https://doi.org/10.1109/RAAI64504.2024.10949519.

[65] R. Sapkota *et al*., "YOLO advances to its genesis: a decadal and comprehensive review of the You Only Look Once (YOLO) series," *Artificial Intelligence Review*, vol. 58, no. 274, 2025, https://doi.org/10.1007/s10462-025-11253-3.