ISSN 2355-3286

One-Phase Smart Switch using OpenCV Hand Gesture Recognition

Mochamad Rizal Fauzan[1], Saiqa Fatur Khairi [1], Neneng Puspita Kaniarudi [1], Silmi Ath Thahirah Al Azhima[2], Nurul Fahmi Arief Hakim[2], Iwan Kustiawan[3], Mariya Al Qibtiya[2], Siscka Elvyanti[1]
[1] Pendidikan Teknik Elektro, UPI, Bandung, Indonesia
[2] Pendidikan Teknik Otomasi Industri dan Robotika, UPI, Bandung, Indonesia
[3] Teknik Elektro, UPI, Bandung, Indonesia
rizalfauzan@upi.edu, saiqafkh@upi.edu, nenengpuspita57@upi.edu, silmithahirah@upi.edu, nurulfahmi@upi.edu, iwan_kustiawan@upi.edu, marialqibtiya@upi.edu, sisckael@upi.edu

Accepted on 26 May 2024
Approved on 07 June 2024

***Abstract*— The need for simplicity in various activities encourages further technological development. One of them is a system to turn lights on and off with just a hand gesture. This hand gesture-based One-phase smart switch uses OpenCV, Arduino Nano, relays, and webcam cameras to recognize hand gestures. Static finger movements are used as buttons to turn on the lights. The results show that the algorithm used has high reliability with a precision score of 0.90, sensitivity of 0.90, accuracy of 0.96, and F1-score of 0.90. The accuracy of the system is affected by light intensity, distance, and hand tilt angle. At a light intensity of 70 LUX, the accuracy reaches 100%, while at 40 LUX the accuracy ranges from 98-99%. A distance of 30-60 cm gave the best accuracy of 100%, but decreased at longer distances. A hand tilt of 0° gives 100% accuracy, while at an angle of 60° the accuracy drops significantly, especially for the fifth finger with 64% accuracy. The average response time of the light to finger movement is 0.133 seconds. This device can recognize a variety of finger patterns well, thus meeting the desired needs.**

***Index Terms*— Arduino; computer vision; hand gesture; OpenCV.**

I. Introduction

Developing machines that can operate in the prompting of humans has been the goal of technological development since its inception. In the early years of development, people used buttons, joysticks, etc. to control circuits, fluid flows, and even mechanical transmissions to fulfill the purpose of the commands given to the machine [1]. When the modern era arrived, with further developments in the world of digital computing, HMI (*Human-Machine Interaction*) environments became even more affordable for many people. People can send information through computers with only a keyboard and mouse in their hands, and a monitor in front of their eyes [2].

In recent technological advancements, researchers have been seeking to simplify their access to HMI. The most suitable approach is a straightforward communication flow. Understandably, there are not many instruments tied to the user's body, and keeping the conversation natural. From then, the interaction system employing hand gestures was restored as a non-verbal communication flow between humans and machines [3].

In the beginning, the concept relied on a greater number of sensors and cameras to accurately record the movement of hands. z, Both static and dynamic gestures. The hand that is modeled must be equipped with a dataglove, which is capable of detecting any alterations that occur in the hand, such as finger movements or finger bent. Despite being published, this invention had a short lifespan due to its reliance on supplementary gear. Hence, employing a camera is considered to be a preferred method for the advancement of a motion tracking system.[4].

The main purpose of using hand motion tracking technology is simply to recognize and classify gestures formed by human hands. These gestures will be captured by the camera, processed by computer algorithms, and learned by the ANN concept to generate program decisions from predetermined gestures [5]. In this paper, the gesture defined is the simple gesture of "bending fingers" as if counting. Each extended finger will be captured by the camera and processed to activate the "turn on the light" program with five different levels according to the five fingers.

II. Methods

A. Computer Vision

Computer vision is a term that broadly covers a variety of areas, including raw data retrieval, image pattern extraction, and information interpretation [6]. The field includes a blend of principles, methodologies, and ideas derived from digital image processing, pattern recognition, artificial intelligence, and computer graphics [7].

Advances in computer vision depend on its supporting infrastructure, which includes improved

ISSN 2355-3286

image quality and image recognition capabilities. Advances in computer vision depend on the computer technology infrastructure, which includes improved image quality and image recognition capabilities [8][9].

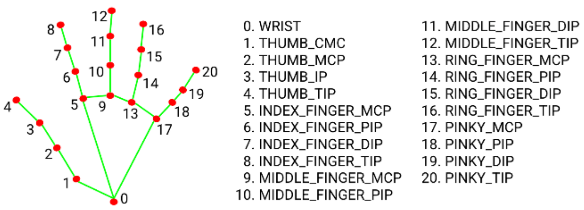
There is a close relationship between Image Processing and basic approaches, and some researchers use the two words interchangeably. The main goal of Computer Vision is to generate models and extract data and information from images. This goal sometimes shares a common meaning and sometimes overlaps with *Human and Machine Interaction* (HMI) [10].

Computer Vision works by using optical sensors and algorithms to recreate human motion stimuli to extract valuable information automatically [11]. When compared to conventional methods, computer vision has evolved into one of the branches of AI for human motion stimulus in society [12].

B. Hand Gesture Recognition

Hand gestures are part of nonverbal communication, which can be expressed through the center area of the palm, the position of the fingers, and the shapes formed by the hands. Hand gestures are divided into two categories, which are static gestures characterized by a stable hand shape, and dynamic gestures involving hand movements such as waving. Hand gestures vary depending on the person and the situation at hand, unlike posture, which emphasizes hand shape rather than movement [13].

Convolutional Neural Network (CNN) is used by neural networks to identify hand gestures using deep learning. CNN or ConvNet, is a type of artificial neural network designed to mimic the pattern of connections found in the visual cortex of objects. CNNs exploit the spatial correlations inherent in the input data. Convolutional networks are used to train the model using a dataset of images. The model is trained using three forms of training: supervised, unsupervised, and reinforcement learning. Given that gesture recognition models rely on supervised learning and derive knowledge from labeled datasets of desired gestures, it is crucial to have high-quality data with little noise. The accuracy of the model network and the data used directly affects the accuracy of the model [14][15].



(a)

(b)

Fig. 1. 21 hand landmarks (a), Hand detection (b)

Research on hand gestures typically uses two main methodologies: sensors integrated into wearable gloves, and sensors based on camera vision. Computerized hand gesture recognition using camera-based systems involves the process of capturing and recognizing hand movements and positions through visual data. This process starts with image capturing through the camera, followed by pre-processing steps such as noise reduction and segmentation to separate the hand image from the noises [16].

C. Hand-Gesture Smart Switch

Hand gesture identification provides a very practical and modern means of non-verbal communication. Gesture recognition has significant applications in Human Computer Interactions (HCI) and sign languages [17]. The hand gesture recognition system utilizes a camera that is primarily used to capture images of gestures made by human hands. These images are then processed by the specified algorithm as input to the created system.

Hand gesture-based controls offer a variety of functions, such as acting as virtual remote controls for TVs and other home appliances, enhancing gaming experiences, and allowing interaction with public information kiosks in museums, ATMs, elevators, storefronts, and other public spaces. At the most recent Consumer Electronics Show (CES), numerous companies presented prototypes or upcoming devices featuring hand gesture-based controls. Different sensing technologies can detect hand gestures, and advancements in these technologies have enabled developers to integrate Three-Dimensional (3D) gesture control into their products. The cost of acquiring and implementing a gesture controller varies depending on the technology employed [18].

III. System Design

The system is designed with four main components: a webcam, an OpenCV program, an Arduino Nano, and a relay. The webcam, whether external or built into the laptop, is used to capture hand gesture images. Then, OpenCV performs real-time image pre-processing to extract only the essential functions from the user's static hand gestures. The system consist on being the control for AC components through the switch function brought by the relays. The switch can be used for many components, but AC lightbulb was used to get the experimental data. These functions are transmitted via a port to the Arduino Nano, which decides which light should be turned on based on the hand gesture. The Arduino Nano then sends a signal to the relay to supply electricity to a specific light according to the captured hand gesture. The sequence of operation of this one-phase smart switch system can be seen in Figure 2.

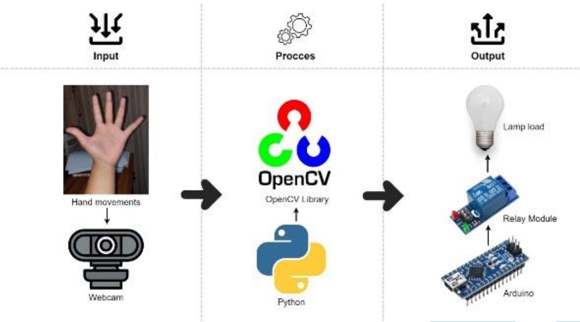
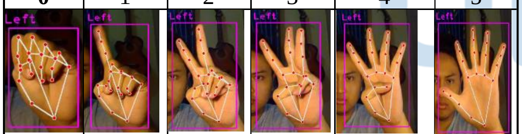


Fig. 2. System Working Stage

Figure 3 shows the classification of hand gestures to control the lighting system. Each hand gesture from 0 to 5 represents the number of fingers raised, with each finger activating one light. A no gesture turns off all relays, while a 5 gesture turns on all relays. Overall, this combination of hand gestures can result in 32 combinations of light control that can be formulated in 2^n.

Classification of Hand Gestures					
0	1	2	3	4	5



Commands are executed on the lighting system					
All Relay Off	Relay 1 ON	Relay 1 and 2 ON	Relay 1, 2 and 3 ON	Relay 1, 2, 3 and 4 ON	All Relay ON

Fig. 3. The style of hand movements and directions used

In Figure 4 the flowchart diagram illustrates the process of controlling relays using hand gesture detection through Arduino. It starts with setting the serial port and connection between the computer and Arduino, then initializing the relay. The Arduino detects hand movements with sensors, and if movement is detected, the system will control the relays according to the finger movement. This process takes place in a master loop until the system shuts down or the command finishes executing.

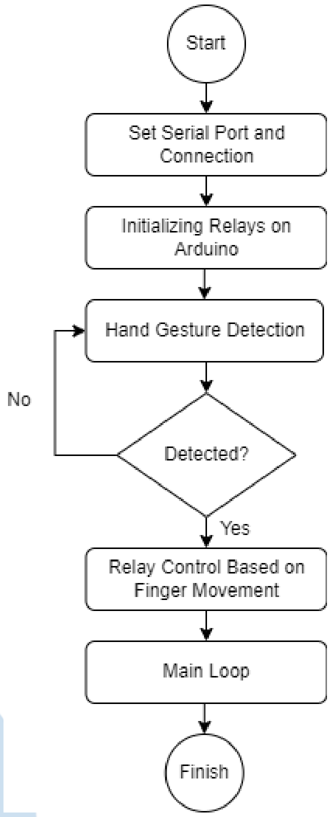


Fig. 4. System Flowchart

In the Arduino Nano system itself, as shown in Figure 5, the components needed are the Arduino Nano as a microcontroller that will turn the lights on or off, a relay that functions as a switch, and five lights as evidence of electricity flowing.

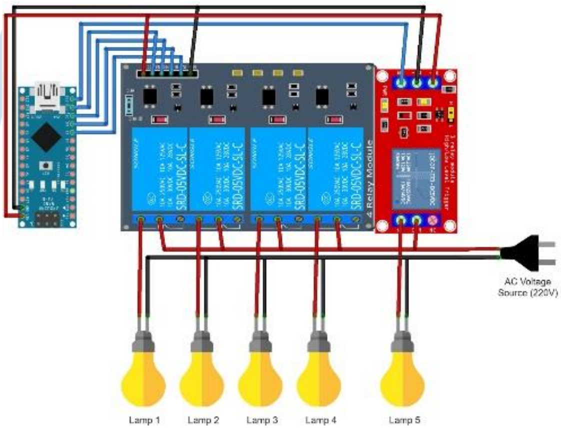
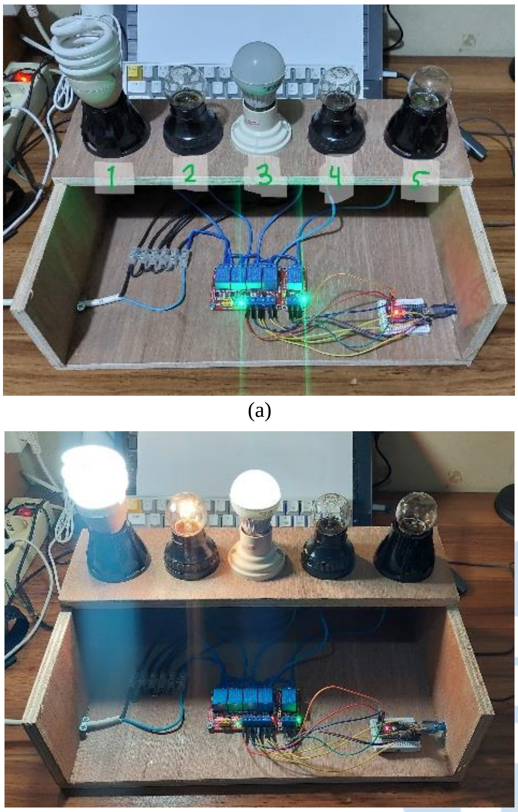


Fig. 5. System Circuits

ISSN 2355-3286

IV. Results and Discussion

A. Hardware Design



(a)

(b)

Fig. 6. Description by lamp number (a), lit condition (b)

The hardware design of the device seen in Figure 6 has a configuration where each lamp number is connected to the corresponding relay, allowing for the control of the lights' on and off states. Specific hand gestures will trigger the relevant relay, causing the bulb associated with that relay number to illuminate. As an illustration, raising a single finger will trigger relay 1 and illuminate lamp 1; raising two fingers will trigger relays 1 and 2, and illuminate lights 1 and 2, and so on. This enables convenient and effective management of the lighting system using manual movements of the hands.

B. Light Activation Logic

The specified static motion is employed to conduct device experiments, which involve the exploration of all feasible patterns. The essential motion for turning on the light, in this case, was the action of "closing and opening the fingers". This gesture serves as a benchmark due to its simplicity and minimal energy expenditure. Data collection is conducted by observing if the lights will illuminate in response to the finger being opened, either individually or in a certain order. Subsequently, the data is acquired and shown in Table 1.

TABLE I. Light Activation Data

Finger Up Combination	**Lamp**				
	1	**2**	**3**	**4**	**5**
0,0,0,0,0					
0,0,0,0,1					Y
0,0,0,1,0				Y	
0,0,0,1,1				Y	Y
0,0,1,0,0			Y		
0,0,1,0,1			Y		Y
0,0,1,1,0			Y	Y	
0,0,1,1,1			Y	Y	Y
0,1,0,0,0		Y			
0,1,0,0,1		Y			Y
0,1,0,1,0		Y		Y	
0,1,0,1,1		Y		Y	Y
0,1,1,0,0		Y	Y		
0,1,1,0,1		Y	Y		Y
0,1,1,1,0		Y	Y	Y	
0,1,1,1,1		Y	Y	Y	Y
1,0,0,0,0	Y				
1,0,0,0,1	Y				Y
1,0,0,1,0	Y			Y	
1,0,0,1,1	Y			Y	Y
1,0,1,0,0	Y		Y		
1,0,1,0,1	Y		Y		Y
1,0,1,1,0	Y		Y	Y	
1,0,1,1,1	Y		Y	Y	Y
1,1,0,0,0	Y	Y			
1,1,0,0,1	Y	Y			Y
1,1,0,1,0	Y	Y		Y	
1,1,0,1,1	Y	Y		Y	Y
1,1,1,0,0	Y	Y	Y		
1,1,1,0,1	Y	Y	Y		Y
1,1,1,1,0	Y	Y	Y	Y	
1,1,1,1,1	Y	Y	Y	Y	Y

Based on the data obtained, the lights can be turned on according to the number and type of fingers raised. If the thumb is raised, then lamp 5 will light up which is symbolized in variable Y. If the index finger and middle finger, then lamp 2 and lamp 3 will light up. This is following the program created.

C. Model Algorithm Accuracy

To achieve high accuracy in detecting and classifying various hand gestures. The performance evaluation of the model was performed by considering several key metrics such as accuracy, recall, precision, and F1 score. In addition, environmental settings such as a distance between the hand gesture and the camera of 40 cm and a tilt angle between the hand gesture and the camera of 0° were determined.

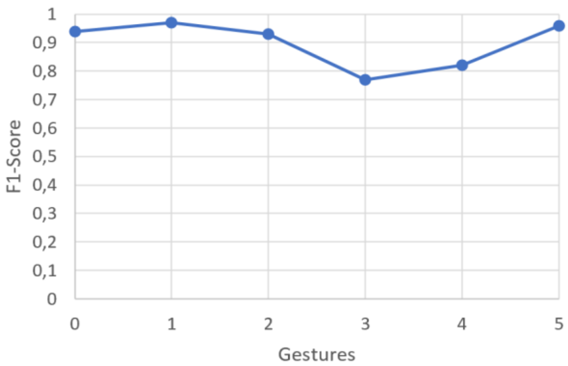


Fig. 7. F1 score results from different classes

The results from Figure 7 show the F1 scores for each class as follows: 0.94, 0.97, 0.93, 0.77, 0.82, and 0.96, with an overall average of 0.898. These scores indicate a consistent and high performance in hand gesture classification, despite variations between different classes. A high F1 score indicates a good balance between precision and recall, reinforcing the reliability of the model in detecting and classifying hand gestures accurately.

TABLE II. CLASSIFICATION REPORT

Class	Precision	Sensitivity	Accuracy	F1-Score
0	0,97	0,92	0,97	0,94
1	0,95	0,96	0,98	0,97
2	0,9	0,97	0,96	0,93
3	0,82	0,74	0,96	0,77
4	0,82	0,84	0,94	0,82
5	0,93	0,98	0,98	0,96
Mean	0,90	0,90	0,96	0,90

The results of Table 2 show that the model has high performance with an average precision of 0.90, sensitivity of 0.90, accuracy of 0.96, and F1 score of 0.90. The performance per class varied, with class 1 having the highest precision of 0.95 and F1 score of 0.97, while class 3 had the lowest F1 score of 0.77. Overall, the model showed good reliability in classifying hand gestures with high accuracy in all classes.

D. Light Activation Logic

Based on the change in lighting intensity, the accuracy of the system is calculated as the first test for the proposed system. A specialized application is used to measure the lighting intensity based on a mobile phone camera. In this study, lighting with variable intensities of 40 lux and 70 lux was used. In addition, environmental settings such as a distance between the hand gesture and the camera of 40 cm and an inclination angle between the hand gesture and the camera of 0° were also determined. The experiment was repeated 10 times for each hand gesture and light intensity.

TABLE III. ACCURACY BASED ON VARIATIONS IN LIGHT INTENSITY

Hand Gesture	Accuracy with Light Intensity (%)	
	40 LUX	**70 LUX**
Fingerless	98	100
One Finger	97	100
Two Fingers	98	100
Three Fingers	98	100
Four Fingers	98	100
Five Fingers	99	100

The results of Table 3 show that the accuracy of the system in detecting hand gestures is very high, with all gestures achieving 98-99% accuracy at 40 lux light intensity and 100% at 70 lux light intensity. This shows that the system functions very well under various lighting conditions.

TABLE IV. ACCURACY BASED ON DISTANCE VARIATION

Hand Gesture	Accuracy with Distance (%)				
	30 cm	**60 cm**	**90 cm**	**150 cm**	**180 cm**
Fingerless	100	100	100	93	82
One Finger	100	100	97	89	78
Two Fingers	100	100	98	88	78
Three Fingers	100	100	99	88	77
Four Fingers	100	100	99	89	78
Five Fingers	100	100	100	91	79

From the results of Table 4 of hand gesture accuracy testing at various distances, it can be concluded that hand gesture recognition has a high level of accuracy at closer distances, but decreases little by little as the distance increases. /Specifically, at a distance of 30 cm, all types of hand gestures have 100% accuracy, but the accuracy starts to decrease at a distance of 60 cm and further decreases at a distance of 90 cm and above. Nonetheless, despite the decrease, accuracy remained relatively high even at longer distances, with the lowest decrease occurring for fingerless gestures.

TABLE V. ACCURACY BASED ON TILT ANGLE VARIATION

Hand Gesture	Accuracy with Tilt Angle (%)		
	0º	**30º**	**60º**
Fingerless	100	98	95
One Finger	100	97	82
Two Fingers	100	98	84
Three Fingers	100	97	72
Four Fingers	100	97	75
Five Fingers	100	100	64

Based on Table 5 of the test results, it can be seen that the accuracy of hand gesture recognition tends to decrease as the tilt angle increases. At an angle of 0°, the accuracy is generally high, but decreases significantly at an angle of 60°. This shows that inclination affects the system's ability to recognize hand gestures, with the lowest accuracy occurring at higher inclination angles.

E. Light Activation Logic

The time test is conducted to find out the elapsed time between the input being provided and the execution of the output. In this scenario, manual gestures serve as inputs that initiate the motion of the light arrangement. The reaction time is determined based on the instant the input is provided. The laptop does data processing and transmits instructions directly to the Arduino Nano over the COM connection. Subsequently, every hand motion is subjected to response time testing. Each manual motion underwent three rounds of testing.

TABLE VI. LAMP RESPONSE TIME TEST

Hand Gesture	Reaction Time Average (s)
Fingerless	0,023
One Finger	0,148
Two Fingers	0,137
Three Fingers	0,192
Four Fingers	0,134
Five Fingers	0,164
Total Average Reaction Time (s)	0,133

The results are shown in Table 6 for testing reaction time to hand movements. The result shows an overall average reaction time of 0.133 seconds. Fingerless hand movements have the fastest reaction time (0.023 seconds), while movements with three fingers require the longest reaction time (0.192 seconds). Movement complexity affects reaction time, with simpler movements tending to have faster responses.

V. CONCLUSION

Based on the analysis conducted in this study, it can be This paper develops a hand gesture-based *One-Phase Smart Switch* using CNN algorithm in OpenCV program library with Arduino Nano, relay, laptop webcam camera, and OpenCV program on *Python platform*. The main goal of this research is to create a device for better HMI communication, especially as a switch to a one-phase AC line. In this research, AC lightbulbs were used as the experimental components, with the main goal that the light can turn on and of based on passive hand gestures. The results show that the algorithm logic model used has high reliability in classifying hand gestures, with a precision score of 0.90, sensitivity of 0.90, accuracy of 0.96, and F1 score of 0.90. Changes in light intensity, distance, and tilt angle affected the accuracy of the system, although not significantly. At a light intensity of 70 LUX, the accuracy reached 100%, while at 40 LUX it ranged from 98-99%. A distance of 30-60 cm gave the best accuracy of 100%, while a distance of 90-150 cm showed a decrease in accuracy, and at 180 cm the accuracy decreased significantly. Hand tilt at an angle of 0° gives 100% accuracy for all fingers, while at angles of 30° and 60° there is a decrease in accuracy, especially at an angle of 60° for finger 5 with 64% accuracy. The average response time of the light-to-finger movement is 0.133 seconds.

REFERENCES

[1] L. Guo, Z. Lu, and L. Yao, "Human-Machine Interaction Sensing Technology Based on Hand Gesture Recognition: A Review," *IEEE Trans. Human-Machine Syst.*, vol. 51, no. 4, pp. 300–309, 2021.

[2] X. Wang and Z. Zhu, "Context understanding in computer vision: A survey," *Comput. Vis. Image Underst.*, vol. 229, p. 103646, Mar. 2023.

[3] E. Ertugrul, P. Li, and B. Sheng, "On Attaining User-friendly Hand Gesture Interfaces to Control Existing GUIs," *Virtual Real. Intell. Hardw.*, vol. 2, no. 2, pp. 153–161, 2020.

[4] H. Sharma, H. Kumar, and S. K. Mangla, "Enablers to computer vision technology for sustainable E-waste management," *J. Clean. Prod.*, vol. 412, p. 137396, Aug. 2023.

[5] F. I. Eyiokur *et al.*, "A survey on computer vision based human analysis in the COVID-19 era," *Image Vis. Comput.*, vol. 130, p. 104610, Feb. 2023.

[6] V. Wiley and T. Lucas, "Computer Vision and Image Processing: A Paper Review," *Int. J. Artif. Intell. Res.*, vol. 2, no. 1, p. 22, 2018.

[7] Shreya M. Shelke, Indrayani S. Pathak, Aniket P. Sangai, Dipali V. Lunge, Kalyani A. Shahale, and Harsha R. Vyawahare, "A Review Paper on Computer Vision," *Int. J. Adv. Res. Sci. Commun. Technol.*, vol. 3, no. 2, pp. 673–677, 2023.

[8] Y. Weiss, V. Ferrari, C. Sminchisescu, and M. Hebert, "Special Issue: Advances in Architectures and Theories for Computer Vision," *Int. J. Comput. Vis.*, vol. 128, no. 3, pp. 573–574, 2020.

[9] H. Yu, Y. Wang, Y. Tian, H. Zhang, W. Zheng, and F. Y. Wang, "Social Vision for Intelligent Vehicles: From Computer Vision to Foundation Vision," *IEEE Trans. Intell. Veh.*, vol. 8, no. 11, pp. 4474–4476, Nov. 2023.

[10] A. Ardanza, A. Moreno, Á. Segura, M. de la Cruz, and D. Aguinaga, "Sustainable and flexible industrial human machine interfaces to support adaptable applications in the Industry 4.0 paradigm," *Int. J. Prod. Res.*, vol. 57, no. 12, pp. 4045–4059, 2019.

[11] A. Batch, Y. Ji, M. Fan, J. Zhao, and N. Elmqvist, "uxSense: Supporting User Experience Analysis with Visualization and Computer Vision," *IEEE Trans. Vis. Comput. Graph.*, 2023.

[12] M. Rafiei, J. Raitoharju, and A. Iosifidis, "Computer Vision on X-Ray Data in Industrial Production and Security Applications: A Comprehensive Survey," *IEEE Access*, vol. 11, pp. 2445–2477, 2023.

[13] M. Yasen and S. Jusoh, "A systematic review on hand gesture recognition techniques, challenges and applications," *PeerJ Comput. Sci.*, vol. 2019, no. 9, pp. 1–30, 2019.

[14] N. Bilal, V. Indradeep, S. Simran, and G. Mansi, "Hand Gesture Recognition," *IJRASET Int. J. Res. Appl. Sci. Eng. Technol.*, vol. 10, no. 5, pp. 5–24, 2020.

[15] A. Magdy, H. Hussein, R. F. Abdel-Kader, and K. A. El Salam, "Performance Enhancement of Skin Cancer

Classification Using Computer Vision," *IEEE Access*, vol. 11, pp. 72120–72133, 2023.

[16] M. Al-Hammadi *et al.*, "Deep learning-based approach for sign language gesture recognition with efficient hand gesture representation," *IEEE Access*, vol. 8, pp. 192527–192542, 2020.

[17] S. P, A. Jain, P. Jaiswal, R. Kumar, and D. Koolwal, "Hand Gesture Recognition System in Smart Environment," *Int. J. Recent Technol. Eng.*, vol. 9, no. 1, pp. 2194–2199, 2020.

[18] V. V. Dahale and D. V. T. Gaikwad, "Hand Gesture Based Touch-Free User Interface for Elevator/Lift," *Int. J. Adv. Res. Sci. Commun. Technol.*, vol. 7, no. 2, pp. 237–241, 2021.