Hand Gesture Recognition and Control for Human-Robot Interaction Using Deep Learning

Philip Jonah Ezigbo[1], Onyebuchi Chikezie Nosiri[2,*], Ekene Samuel Mbonu[1], Victor Ofor[1], and Jude Obichere[1]
[1] Department of Mechatronics Engineering, Federal University of Technology Owerri, Nigeria
[2] Department of Telecommunication Engineering, Federal University of Technology Owerri, Nigeria
Email: philip.ezigbo@futo.edu.ng (P.J.E.), onyebuchi.nosiri@futo.edu.ng (N.O.C.), ekene.mbonu@futo.edu.ng (E.S.M.), oforchukwuebuka.20171040373@futo.edu.ng (V.O.), jude.obichere@futo.edu.ng (J.O.)

***Abstract*—This paper introduces a real-time system for recognizing hand gestures using Python and OpenCV, centred on a Convolutional Neural Network (CNN) model. The primary objective of this study is to address the challenge of recognizing hand gestures in varied and complex environments. The proposed approach employs several image and video processing techniques, including data augmentation and feature extraction, to segment the hand region and extract relevant features. The system's performance is significantly improved by adding to the original training dataset, resulting in 5,000 images with 500 images per gesture, as shown by the evaluation metrics indicating a substantial increase in accuracy from 96.9% to 99.2%. This paper aims to provide feasible and economical solutions for utilizing robots in industrial settings, while also proposing future research possibilities for enhancing human-robot interaction through methods such as incorporating hand gesture recognition.**

***Index Terms*—Convolutional neural networks, computer vision, deep learning, hand gesture recognition, human-robot interaction**

I. Introduction

Hand Gesture Recognition (HGR), has grown in popularity in recent times as an interesting research area. It has reformed the manner of human-computer interaction by allowing natural and spontaneous communication through hand movements. One such application involves facilitating communication for disabled individuals by translating hand signs into spoken words, thereby bridging the communication gap [1]. In the field of medical rehabilitation, hand gesture recognition plays a crucial role in monitoring and evaluating hand movements, aiding in the recovery process [2]. It contributes to the enhancement of gaming and augmented reality experiences by providing natural and intuitive gesture control, which results in more immersive interactions [3]. In daily life, this technology simplifies the management of home appliances, allowing users to effortlessly operate those using gestures such as tapping, swiping, or rotating [4]. In the field of industrial automation, hand gesture recognition enables precise and efficient control of robots and Unmanned Aerial Vehicles (UAVs) through gestures, thereby streamlining various industrial processes [5, 6].

Various sensors, such as gloves [7], joysticks [8], electromyography (EMG) signals [9], inertial measurement units (IMU) [10], and cameras [11], could be employed for HGR. Camera-based HGR has gained significant popularity among these sensors due to its non-intrusive nature and lack of physical contact. This approach utilizes cameras to capture hand movements without any external disruption. However, camera-based HGR is faced with challenges such as occlusion, variations in illumination, background interference, and changes in hand positioning, hence, necessitating ongoing research and improvement.

In the early stages of hand gesture recognition research, pioneers devised different methodologies to overcome the obstacles associated with hand detection and tracking [12]. Two notable approaches emerged: data gloves and marker-based methods [13, 14]. Data gloves are wearable devices equipped with sensors, including flex sensors, accelerometers, and gyroscopes, enabling precise measurement of hand articulation and orientation. Marker-based methods involved affixing markers or fiducial markers onto the user's hands, which were subsequently tracked by cameras or optical sensors. By monitoring the marker's position and movement, these techniques facilitated the accurate determination of hand gestures. Despite their effectiveness, these approaches required additional equipment and often proved uncomfortable for users. Consequently, early research efforts paved the way for the development of more convenient and non-intrusive vision-based gesture recognition systems. These advanced systems utilized cameras to detect and track hand movements without the need for wearables or markers, marking a significant milestone in HGR innovation.

Robot Control based on hand gesture recognition has become particularly important in industrial hazardous environments such as mining sites, nuclear plants, and chemical reaction facilities, where operators need a safer and more efficient way to control machines. Recognizing hand gestures in these hazardous environments, pose a

Manuscript received June 3, 2023; revised August 7, 2023; accepted August 12, 2023.
*Corresponding author

significant challenge due to the complexity and unpredictability of such environments. These robotic systems must operate in challenging conditions like poor lighting, cluttered spaces, and sudden changes in temperature, pressure, and humidity, creating a difficult scenario in developing a robust hand gesture recognition system that can withstand unpredictable events. Conversely, hand gestures could as well be occluded by other objects or body parts, affected by varying lighting conditions, background clutter, or distorted by camera noise or motion blur. On this premise, the paper seeks to address the issue by enhancing the accuracy of the existing Convolutional Neural Network (CNN) model using diverse hand gesture datasets from American Sign Language (ASL) which contains 700 images from 5 individuals, with variations in lighting conditions and hand postures

II. Literature Review

Smart gloves have received considerable attention as a plausible solution for interfaces involving vision and voice interaction. However, their practical implementation is frequently impeded by the trade-off between functionality, performance and cost due to limitations and inaccuracy of hand gesture recognition, both in static and dynamic gestures. In an effort to address these challenges, Gu *et al*. [15] proposed a wireless smart glove-based interface. Their approach utilized an all-recyclable, ultra-stretchable sensing fibre composed of liquid metal and thermoplastic materials. This innovation enabled highly accurate static and dynamic hand gesture recognition by ensuring high skin compliance and scalability.

Camera-based HGR approaches often face challenges associated with noise impact, gesture feature extraction, and the utilization of continuous gesture time sequential information. Addressing these issues, Wang *et al*. [16] proposed a time sequential inflated 3 dimensions (TS-I3D) convolutional neural network approach for HGR, utilizing Frequency Modulated Continuous Wave (FMCW) radar sensors. Their method effectively extracted range and speed change information from Range-Doppler Maps (RDMs) generated by the FMCW radar, leading to a high average recognition accuracy rate.

Static HGR, whether user-dependent or user-independent, can be particularly challenging, especially in scenarios involving lighting changes, hand position variations, and complex backgrounds. To overcome these difficulties, authors of [17] proposed a recognition method that leverages image descriptors such as Gradient Local Auto-Correlation (GLAC), Gabor Wavelet Transform (GWT), and Fast Discrete Curve Transform (FDCT). Dimensionality reduction through Principal Component Analysis (PCA) further enhances their approach. Remarkably, their study achieved exceptional results, with 100% accuracy for user-independent gestures and 98.33% accuracy for user-dependent gestures.

To overcome on-site environmental disturbances such as poor illumination, fog, and dust affecting hand gesture recognition, the authors of [18] focused on thermal image-based hand gesture recognition for worker-robot collaboration in the construction industry. Their experimental results indicated that thermal images demonstrate robustness under different lighting conditions.

The use of Recurrent Neural Networks (RNNs) is particularly advantageous in the analysis of hand gestures represented as sets of feature vectors that change over time. Avola *et al*. [19] took advantage of this feature and applied RNN to model the contextual information that is embedded in the temporal sequence of hand gesture feature vectors. Their method involved capturing finger bone angles using a leap motion controller sensor, and it achieved remarkable accuracy, exceeding 96% when tested on a challenging dataset of American Sign Language gestures.

Traditional RNNs may face difficulties in recognizing dynamic gestures due to their restricted capacity for processing data in a single direction. To address this constraint, Lin *et al*. [20] introduced a gesture recognition technique and device that exploit light sensing characteristics. By utilizing the photoelectric sensing capability of LED screens, their proposed method eliminated the need for external sensors. The system incorporated Field-Programmable Gate Array (FPGA) control and deep learning analysis, employing static bidirectional long short-term memory (S-Bi-LSTM) for static gestures and an optimized dynamic bidirectional long short-term memory (D-Bi-LSTM) for dynamic gestures. Experimental results demonstrated remarkable accuracy for dynamic gestures.

An intriguing area of research in HGR is the use of Electrical Impedance Tomography (EIT) to analyze impedance changes within the arm, allowing for the inference of muscle contractions. In [21], Li *et al*. developed a system that utilized EIT to detect muscle contractions and recognize hand signs. This system encompasses an electronic interface, an image reconstruction algorithm, a CNN classifier, and a virtual hand model. Impressively, their approach achieved high accuracy in recognizing American Sign Language (ASL) numbers, surpassing the performance of the Support Vector Machine (SVM) classifier. Table I shows the recognition accuracy comparison with the conventional studies.

TABLE I: RECOGNITION ACCURACY COMPARISON WITH PREVIOUS STUDIES

Method	Accuracy Result	Limitation
Wireless smart glove-based interface combined with ML and self-adaptive algorithm [15].	Only 11 hand gestures were considered with accuracy of 93.6%	Limited Datasets

Time Sequential Inflated 3 Dimensions (TS-I3D) CNN approach for HGR based on Frequency Modulated Continuous Wave (FMCW) radar sensors [16].	Their experimental results showed average recognition accuracy rate of 96.17%.	Limited Datasets and work focused only on gesture feature extraction and the impact of noise on hand gesture parametric images.
static hand gesture recognition based on a set of image descriptors: Gradient Local Autocorrelation (GLAC), Gabor Wavelet Transform (GWT), and Fast Discrete Curve Transform (FDCT) [17].	with 100% accuracy for user-independent gestures and 98.33% accuracy for user-dependent gestures	Limited datasets (the study investigated gesture recognition by a single user and different users, utilizing three datasets for user-independent recognition and one for user-dependent recognition).
Thermal image-based hand gesture recognition [18].	Experimental results indicated that thermal images were robust to different lighting conditions, and the proposed model achieved a high classification accuracy of 97.54 % with 1.8 M parameters.	The model focuses mainly on computational efficiency than accuracy of gesture recognition.
Recurrent neural networks (RNNs) to model the long-term contextual information inherent in temporal sequences of hand gesture feature vectors [19].	Gesture recognition accuracy of 96%	Involved large number gestures defined by The American Sign language and semaphoric hand gestures gotten from Shape Retrieval Contest datasets.
Gesture recognition method and device exploiting light sensing characteristics, using FPGA control and deep learning analysis, employing S-Bi-LSTM and D-Bi-LSTM [20].	Remarkable accuracy of 91.67% for dynamic gestures	Limited Datasets
CNN and OpenCV	Significant improvement on original training datasets resulting in 5000 images with 500 images per gesture. And Improved recognition accuracy of 99.2% despite large datasets.	Large and Augmented Datasets

In summary, these reviewed studies have contributed to the advancement of hand gesture recognition by proposing innovative techniques and achieving high recognition accuracy across various challenges, such as trade-offs in smart gloves, and noise impact in camera-based approaches. However, their reliance on small datasets limits their ability to represent the complexity of hand gestures encountered in real-world settings. To overcome this limitation, this study aims to gather and pre-process a diverse collection of hand gesture images to enhance the overall performance and generalize to new scenarios. Furthermore, the study will utilize image augmentation techniques to improve its own model robustness in challenging environments and leverage Python and OpenCV for efficient and accurate recognition of hand gestures.

III. Proposed System Design

Deploying a high-granularity recognition (HGR) system in a robotic arm with six degrees of freedom entails the utilization of vision sensors, or cameras, to capture the gestures of the operator's hand. The image data is subsequently processed by OpenCV to govern the movements of the robotic arm. The processed images are subjected to a CNN classifier model, which identifies and classifies the gestures, subsequently converting them into commands. This novel approach provides operators with the ability to manipulate the robotic arm using hand gestures instead of a conventional controller. Fig. 1 represents a block diagram of the proposed system design.

A detailed description of the system block diagram is highlighted as follows.

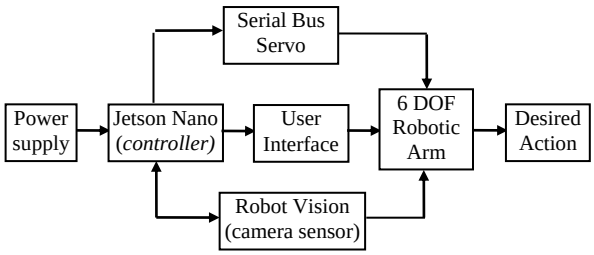

Fig. 1. Block diagram of the hand gesture-controlled industrial robot.

Power Supply: This unit provides the power that is needed to run the entire system. The Jetson nano controller will be powered by a 5V, 4A power adapter. The adapter is characterized by short circuit protection and overload protection.

Controller: The HGR system controller is implemented with an Nvidia Jetson Nano microcomputer [22], specifically created for AI and robotic purposes, presenting a high-performance GPU and quad-core ARM Cortex-A57 CPU to execute intricate algorithms and machine learning models. It undertakes critical tasks such as image processing, object recognition, and gesture recognition, offering the necessary framework to operate a robotic arm's control algorithms and hand gesture recognition software.

Serial Bus Servo: Serial bus servos are motorized actuators that possess the ability to be regulated via a serial communication protocol. Due to their precise positioning and control capabilities, they are fitting for the suggested design. The six degrees of freedom (DOF) robotic arm necessitates these various servos to moderate its joints. The utilization of serial bus servos simplifies the process of wiring and controlling multiple servos as

they can be connected in a chain and individually or collectively governed.

Robot Vision: Robot vision, typically implemented using cameras, enables the robot to perceive its surroundings, recognize objects, and track hand gestures. It captures visual information that can be processed by the AI algorithms running on the Jetson Nano.

DOF Robotic Arm: A robotic manipulator possessing six degrees of freedom has the capacity to move and rotate in various directions. It comprises six joints that utilize a servo motor to facilitate motion. The manipulator is equipped with an end effector, such as a laser, cutter, or gripper that is attached to the end to permit efficient interaction with the surrounding workspace. The choice of a suitable end effector is contingent upon the specific requirements of the task, guaranteeing optimal functionality and performance.

Mobile App Interface: A mobile app remote control is simply an application that can be used to control the robotic arm from a mobile device. It allows users to send commands, control the arm's movements, and potentially even perform hand gesture recognition through their mobile phones.

IV. Methodology

Developing a system for HGR involves two key aspects, namely model development, and deployment. The former necessitates a series of stages such as data preparation, image processing, defining the model architecture, assessing its performance, and conducting tests. The latter phase encompasses the implementation of the model through OpenCV, initializing the camera, extracting hand features, classifying the gestures, and utilizing the recognized gestures to manage external devices via Jetson Nano.

A. Dataset Preparation

The dataset of the hand gestures comprises American Sign Language (ASL) hand gestures as shown in Fig. 2 [23], comprising 700 images from 5 individuals. These images, feature variations in lighting conditions and hand postures, which were achieved through the application of image processing techniques. The hand images possess a single channel and have dimensions of 400 by 400 pixels, with the hands located centrally. To expedite computations and enhance the speed of hand gesture recognition by the system, the images have been scaled down to (64, 64, 1).

The American Sign Language (ASL) employs a set of hand gestures that can be classified as 10 digits, ranging from 0 to 9. Each of these classes comprises 700 images, and the dataset is partitioned into two subsets - one for training (80% of the data) and the other for testing (20% of the data). Furthermore, the number of images per class is equally distributed across both subsets.

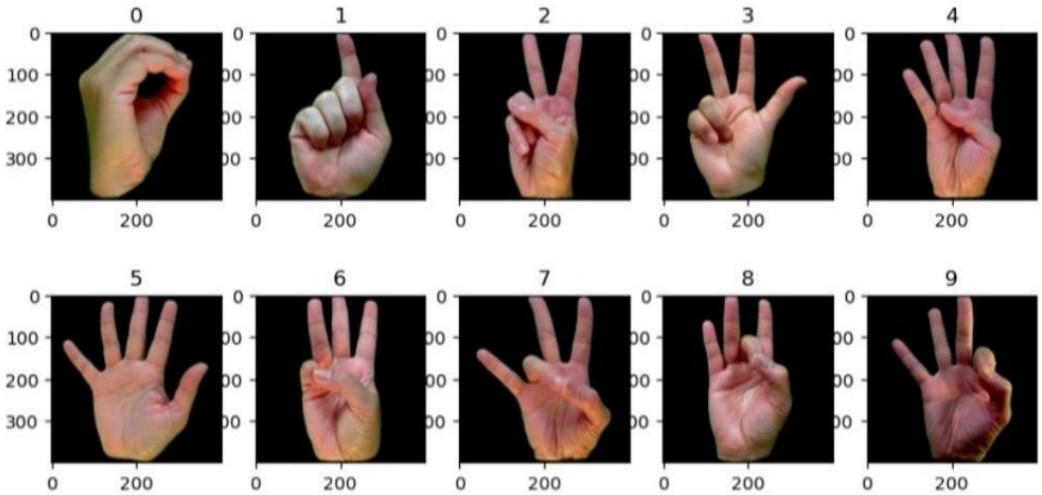

Fig. 2. ASL hand gestures.

B. Data Augmentation

In order to expand the available training data and improve the model's ability to adapt to various lighting conditions and busy backgrounds, several image augmentation techniques were applied to the dataset. These techniques encompassed horizontal shifts, rotations, flips, brightness adjustments, and more [24]. As a result, the dataset size significantly increased from 700 original images to 5,000 augmented images, with an equal distribution of 500 images per set of 10 distinct hand gestures.

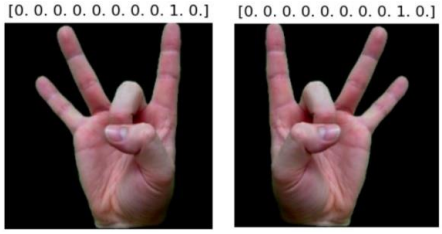

Fig. 3. Image flipped horizontally.

C. Model Classification

A model for gesture recognition tasks, known as Deep Convolutional Neural Network (DCNN), was developed.

The model architecture as depicted in Fig. 4 comprises multiple layers of convolutional, pooling, dropout, and fully connected layers to facilitate classification.

The architecture is specifically designed to recognize hand gestures from input images. The network applies various layers, designed to sequentially extract features and ultimately yield a probability distribution over ten hand gesture classes.

Initial layers of the CNN implement convolutional filters to extract spatial features from the input image, converting these into a feature map. Subsequent layers include max-pooling layers, which decrease the feature map's dimensions and dropout layers, which avert overfitting by randomly omitting units during the training process [25].

The incorporation of non-linearity into the system, crucial for recognizing complex patterns, is achieved through the rectified linear unit (ReLU) activation function applied after each convolutional layer [26]

Mathematically, the ReLU function is given as:

$$f(x) = \max(0, x) = \begin{cases} x \; if \; x > 0, \\ 0 \; otherwise \end{cases} \quad (1)$$

In the final layer of the CNN, SoftMax activation was applied, which transforms the output logits into probabilities. This corresponds to the likelihood of the input image belonging to each of the ten potential hand gesture classes. Importantly, SoftMax activation ensures that the predicted probabilities sum to one, allowing the model to make confident class predictions based on the input image features. The SoftMax function could be expressed as:

$$S(z_k) = \frac{e^{z_k}}{\sum_{j=1}^{K} e^{z_j}} \quad (2)$$

Where z_k is the kth element of the input vector and k is the number of classes in this case k=10. The denominator (canonical partition function) is a normalizing constant to make sure the probabilities add up to unity.

The model adjusts its parameters using root mean square propagation (RMSProp), which guides the model's parameters along the steepest descent. The categorical cross-entropy loss function is implemented to minimize the disparity between the predicted probability distribution ($p = [p_1, p_2, \ldots, p_{10}]$) over the ten hand gestures and the true class labels ($y = [y_1, y_2, \ldots, y_{10}]$).

For a single training example, the categorical cross entropy loss can be calculated as:

$$L = -\sum_{i=1}^{10} y_i \bullet \log(p_i) \quad (3)$$

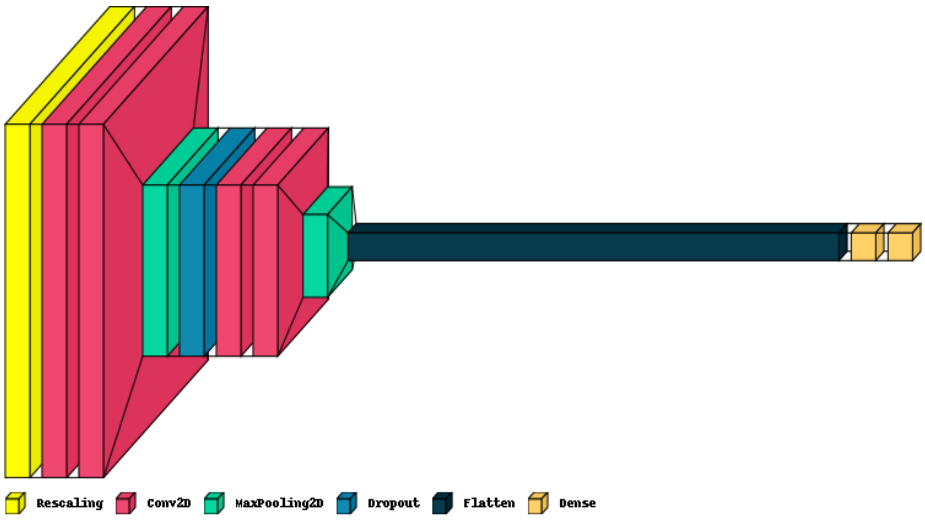

Fig. 4. Model architecture.

In post-training, the model's performance is evaluated using a testing dataset, reporting the accuracy metric and plotting a confusion matrix for the visual representation of the classification results.

The architecture of the CNN model for image classification, illustrated in Fig. 4, is characterized as follows: input images of size 400 × 400 pixels are accepted and subjected to initial rescaling to 64 × 64 pixels. The data is then processed through two Conv2D layers, each with 32 filters and a kernel size of (3, 3), followed by a max-pooling layer and a dropout layer to prevent overfitting while preserving the input shape. Additional Conv2D layers with 64 filters and a max-pooling layer are employed to further process the data. The resulting features are flattened from a 4D tensor to a 2D tensor and fed into two fully connected dense layers. The first dense layer comprises 64 neurons and is connected to the flattened output, while the second layer comprises ten neurons, indicative of the number of classes in the task.

The model, in its entirety, encompasses a total of 987,882 trainable parameters, which comprise the convolutional and dense layers' weights and biases. These parameters are acquired through training and are utilized to enhance the model's efficacy in accomplishing the classification task at hand.

D. Model Development

The model that has been trained is put into action on an HGR system in real-time through the utilization of OpenCV. This HGR system captures video frames using a camera and implements the same pre-processing

measures as were utilized during training. The model then proceeds to make predictions regarding hand gestures in actual time.

E. System Overview

The Python-based hand gesture recognition system that leverages OpenCV and a pre-trained CNN model to detect and classify hand gestures in real time was implemented. Specifically, the system imports necessary libraries, initializes the camera, and creates a gesture recognizer object with the model path and other options. Additionally, the system captures a frame, converts it to RGB format, locates the region of interest (ROI) where the hand is located, and extracts its features, including landmarks, contours, hulls, and so forth. Subsequently, the system feeds these features to the CNN model to obtain the gesture class and confidence score. Based on the gesture class, the system controls a robot and displays the corresponding result on the screen. Lastly, the system iterates through the aforementioned steps until the user terminates the program or presses a key. The steps are illustrated in the developed HGR algorithm.

F. The HGR Algorithm

1. Import the required libraries
2. Load the trained CNN model
3. Initialize the Jetson Nano camera or webcam
4. Set the image dimensions for the model input
5. Define the labels for hand gesture classes
6. Initialize the 6 DOF arm control interface
7. While True:
   1) Capture frame from the camera
   2) Preprocess the frame for hand gesture recognition
   3) Perform hand gesture recognition classification using the CNN model
   4) Get the predicted gesture label
   5) Translate the recognized gesture into a command for the 6 DOF arm
   6) Send the command to the 6 DOF arm control interface
   7) Display the frame and recognized gesture label
   8) Check for the 'q' key press to exit the loop
8. Release the camera and close the OpenCV windows

V. Results and Discussion

The use of hand gesture recognition for robotic control in hazardous environments by leveraging the power of Python and OpenCV was explored in the study. Through a series of experiments and evaluations, valuable insights were obtained into the effectiveness of the approach. The confusion matrix, precision, recall, and F_1 score of the system provided valuable insights into the classification performance of the model. A confusion matrix is a tabular representation that provides an overview of the performance of a machine-learning model with respect to a given test dataset. It summarizes the number of accurate and inaccurate predictions made by a classifier. On the other hand, a confusion matrix heatmap is a graphical depiction of the confusion matrix. It is a two-dimensional matrix that is colour-coded and displays the number of true positives, false positives, true negatives, and false negatives for each class in the classification model. Fig. 5 illustrates the confusion matrix heat map of the test dataset.

The presented visual of the confusion matrix in Fig. 5 depicts a visual portrayal of the confusion matrix, which presents the efficacy of a hand gesture recognition system that utilizes ten (10) distinct features to classify gestures ranging from 0 to 9. Each row in the matrix corresponds to the actual classes, while each column represents the predicted classes. The figures within the matrix denote the frequency or count of predictions made by the system. To expound further, the first row of the matrix illustrates that the system accurately predicted Class 0 for one instance. However, it made errors by predicting Class 1 for two instances, Class 2 for one instance, and so forth. Similarly, the second row of the matrix shows that the system failed to predict Class 1 accurately for any instance. Instead, it misclassified by predicting Class 0 for two instances, Class 2 for one instance, Class 3 for one instance, Class 4 for three instances, and so on. The diagonal elements of the confusion matrix, which run from the top-left to the bottom-right, signify the correct predictions, whereas the off-diagonal elements indicate misclassifications or errors made by the system. An analysis of the confusion matrix provides valuable insights into the performance of the classification model. It was done in order to identify the model's strengths and weaknesses, which can help to make informed improvements or adjustments to enhance its predictive capabilities.

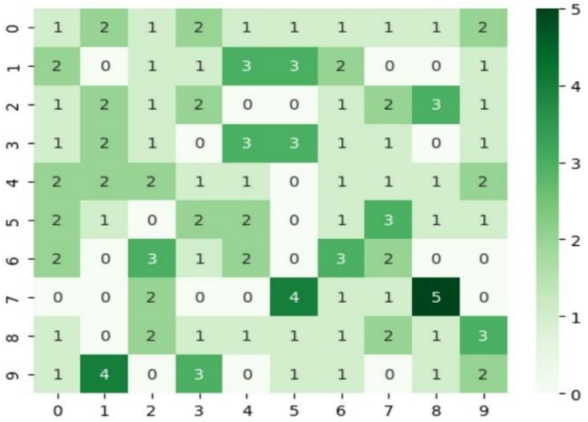

Fig. 5. Confusion matrix heat map of the test dataset.

A. Evaluation Metrics

Accuracy, precision, recall, and F_1 score are performance evaluation metrics that are commonly employed to assess the efficacy of a model. Accuracy gauges the ratio of accurately predicted observations to the overall number of observations. Precision measures the degree to which a model accurately predicts positive observations. Recall, on the other hand, assesses the ability of the model to correctly predict all feasible positive observations. The F_1 score is a composite metric

that computes the weighted average of precision and recall.

$$Accuracy = \frac{T_p+T_n}{T_p+T_n+F_p+F_n} \quad (4)$$

$$Precision = \frac{T_p}{T_p+F_p} \quad (5)$$

$$Recall = \frac{T_p}{T_p+T_n} \quad (6)$$

$$F_1\ score = 2 \times \frac{precision \times recall}{precision+recall} \quad (7)$$

The variables T_p, T_n, F_p, and F_n represent true positives, true negatives, false positives, and false negatives, respectively. The machine learning algorithm was trained using a specific set of training data and evaluated using these metrics on test data. In an effort to increase the amount of training data, the original training dataset was expanded to include a total of 5,000 images, with 500 images corresponding to each gesture. Following training using the augmented dataset and evaluation using the same test data, the model exhibited a significant improvement in performance. Table II shows the evaluation summary.

TABLE II: EVALUATION SUMMARY

Evaluation Metrics of Classifier	Trained with original set	Trained with augmented set
Accuracy	96.9%	99.02%
Precision	90.6%	91.45%
Recall	90.6%	91.45%

Moreover, the analysis presented a method of optimization aimed at reducing training losses through the modification of neural network attributes such as weights and learning rate. An epoch which signifies a complete traversal through the entire training dataset within the model training process was carried out. In essence, the training process is compartmentalized into a specific number of epochs, wherein the model scrutinizes all the training instances once and updates its parameters, namely weights and biases, on the basis of the errors it incurs while predicting the targets. The count of epochs employed during training constitutes a hyperparameter that necessitates tuning in order to realize optimal performance. The deployment of insufficient epochs may lead to underfitting, wherein the model is unable to apprehend the patterns present in the data, whereas an excessive epoch count may culminate in overfitting, where the model becomes too intricate and assimilates noise from the data. Usually, the epoch count is established by striking a balance between the training time and the performance on a validation dataset. Monitoring the validation loss throughout the training process is a customary practice to determine the appropriate point to conclude the training. If the validation loss stops decreasing, it may suggest that the model has achieved its optimal performance, and further training may lead to overfitting. Fig. 6 and Fig. 7 depict the validity accuracy in relation to consistency and loss of validity as contrasted to loss of training. It demonstrates the accuracy and loss epoch for the model.

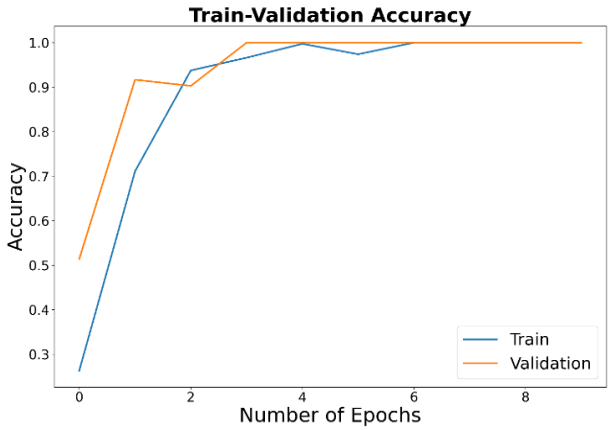

Fig 6. Accuracy vs number of epochs.

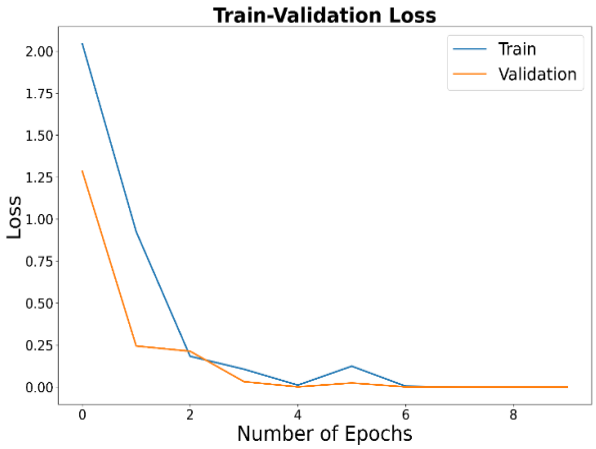

Fig. 7. Loss vs number of epochs.

VI. CONCLUSION

The use of hand gesture recognition for robotic control in hazardous environments by leveraging the power of Python and OpenCV was explored in the study. The study demonstrated that it is possible to design a natural and intuitive interface for humans to interact with robotic systems using these open-source tools. From the results, it can be seen that hand gesture recognition can be performed with high accuracy and reliability using Convolutional Neural Network (CNN) models, especially when they are trained with a varied and representative dataset. From the evaluation summary in Table II, there was a significant increase in accuracy and precision from 96.9% to 99.2% and from 90.6% to approximately 92% respectively.

Additionally, a potential avenue for progress in this research would involve incorporating hand gesture recognition with other forms of control, such as voice and eye-tracking. Specifically, this would entail utilizing hand gestures to identify and direct attention to objects, issuing commands through voice, and monitoring robot actions through eye-tracking. The implementation of this multimodal interface has the potential to optimize the efficiency and efficacy of human-robot interaction, while also bolstering user satisfaction and trust in the robotic system.

CONFLICT OF INTEREST

The authors declare no conflict of interest.

AUTHOR CONTRIBUTIONS

Philip J. Ezigbo developed a Convolutional Neural Network (CNN) model deployed for the hand gesture recognition tasks and also evaluated the model's performance using metrics, demonstrating significant improvements in accuracy and other measures after training with the augmented dataset. Onyebuchi Nosiri was chiefly involved in the critical analysis, evaluation and formatting of the entire work including the review of the manuscript. Victor Ofor played a pivotal role in sourcing and preparing the dataset used in this work and also ensured it was suitable for training purposes while Ekene Mbonu and Jude Obichere assisted in the review, simulation and in formatting of the work.

ACKNOWLEDGEMENT

The authors wish to thank Federal University of Technology, Owerri for providing the enabling environment to carry out the study.

REFERENCES

[1] Y. Obi, K. S. Claudio, V. M. Budiman, S. Achmad, and A. Kurniawan, "Sign language recognition system for communicating to people with disabilities," *Procedia Computer Science*, vol. 216, pp. 13–20, 2023.

[2] B. A. Cruz-Sánchez, M. Arias-Montiel, and E. Lugo-González, "EMG-controlled hand exoskeleton for assisted bilateral rehabilitation," *Biocybernetics and Biomedical Engineering*, vol. 42, no. 2, pp. 596–614, 2022.

[3] W. Fang and J. Hong, "Bare-hand gesture occlusion-aware interactive augmented reality assembly," *Journal of Manufacturing Systems*, vol. 65, pp. 169–179, Oct. 2022.

[4] C.-Y. Yang, Y.-N. Lin, S.-K. Wang, V. R. L. Shen, Y.-C. Tung, F. H. C. Shen, and C.-H. Huang, "Smart control of home appliances using hand gesture recognition in an IoT-enabled system," *Applied Artificial Intelligence*, vol. 37, no. 1, 2023, DOI: 10.1080/08839514.2023.2176607.

[5] A. Saxena, A. Gupta, Z. Mohsin, A. Singh, H. Raghuwanshi, and Y. Singh, "An optimal gesture controlling of the robotic system," *Materials Today: Proceedings*, vol. 79, pt. 2, pp. 398–405, 2023.

[6] S. Togo and H. Ukida, "UAV manipulation by hand gesture recognition," *SICE Journal of Control, Measurement, and System Integration*, vol. 15, no. 2, pp. 145–161, 2022.

[7] F. Camastra and Domenico De Felice, "LVQ-based hand gesture recognition using a data glove," *Neural Nets and Surroundings*, vol. 19, pp. 159–168, Jan. 2013.

[8] S. Bordoni and G. Tang, "Development and assessment of a contactless 3D joystick approach to industrial manipulator gesture control," *International Journal of Industrial Ergonomics*, vol. 93, #103376, 2023, doi: 10.1016/j.ergon.2022.103376.

[9] A. G. Jaramillo and M. E. Benalcázar, "Real-time hand gesture recognition with EMG using machine learning," in *Proc. 2017 IEEE Second Ecuador Technical Chapters Meeting (ETCM)*, Salinas, Ecuador, 2017, doi: 10.1109/ETCM.2017.8247487.

[10] J. Lopes, M. Simão, N. Mendes, M. Safeea, J. Afonso and P. Neto, "Hand/arm gesture segmentation by motion using IMU and EMG sensing," *Procedia Manufacturing*, vol. 11, pp. 107–113, 2017.

[11] A. Singh, V. Kalaichelvi, R. Karthikeyan, and A. H. Darwish, "A survey on vision-guided robotic systems with intelligent control strategies for autonomous tasks," *Cogent Engineering*, vol. 9, no. 1, 2022, doi: 10.1080/23311916.2022.2050020.

[12] P. Premaratne, "Historical development of hand gesture recognition," in *Human-Computer Interaction Using Hand Gestures*, Springer, 2014, pp. 5–29, doi: https://doi.org/10.1007/978-981-4585-69-9_2.

[13] H. Ishiyama and S. Kurabayashi, "Monochrome glove: A robust real-time hand gesture recognition method by using a fabric glove with design of structured markers," in *Proc. 2016 IEEE Virtual Reality (VR)*, Greenville, USA, 2016, pp. 187–188.

[14] L. Guo, Z. Lu and L. Yao, "Human-machine interaction sensing technology based on hand gesture recognition: A review," *IEEE Trans. on Human-Machine Systems*, vol. 51, no. 4, pp. 300–309, Aug. 2021.

[15] W. Gu, S. Yan, J. Xiong, Y. Li, Q. Zhang, K. Li, C. Hou, and H. Wang, "Wireless smart gloves with ultra-stable and all-recyclable liquid metal-based sensing fibres for hand gesture recognition," *Chemical Engineering Journal*, vol. 460, p. 141777, 2023.

[16] Y. Wang, S. Wang, M. Zhou, Q. Jiang, and Z. Tian, "TS-I3D based hand gesture recognition method with radar sensor," *IEEE Access*, vol. 7, pp. 22902–22913, 2019.

[17] K. Sadeddine, F. Z. Chelali, R. Djeradi, A. Djeradi, and S. Benabderrahmane, "Recognition of user-dependent and independent static hand gestures: Application to sign language," *Journal of Visual Communication and Image Representation*, vol. 79, #103193, 2021, https://doi.org/10.1016/j.jvcir.2021.103193.

[18] H. Wu, H. Li, H.-L. Chi, Z. Peng, S. Chang, and Y. Wu, "Thermal image-based hand gesture recognition for worker-robot collaboration in the construction industry: A feasible study," *Advanced Engineering Informatics*, vol. 56, #101939, 2023, doi: 10.1016/j.aei.2023.101939.

[19] D. Avola, M. Bernardi, L. Cinque, G. L. Foresti, and C. Massaroni, "Exploiting recurrent neural networks and a leap motion controller for the recognition of sign language and Semaphoric hand gestures," *IEEE Transactions on Multimedia*, vol. 21, no. 1, pp. 234–245, Jan. 2019.

[20] P. Lin, R. Zhuo, S. Wang, Z. Wu, and J. Huangfu, "LED screen-based intelligent hand gesture recognition system," *IEEE Sensors Journal*, vol. 22, no. 24, pp. 24439–24448, Dec. 15, 2022.

[21] X. Li, J. Sun, Q. Wang, R. Zhang, X. Duan, Y. Sun, and J. Wang., "Dynamic hand gesture recognition using electrical impedance tomography," *Sensors*, vol. 22, no. 19, #7185, Sep. 2022.

[22] NVIDIA Corporation. Jetson Nano System-on-Module Data Sheet. NVIDIA Developer, NVIDIA Corporation, 2022. [Online]. Available: https://developer.nvidia.com/downloads/embedded/dlc/jetson-nano-system-module-datasheet.

[23] V. Ofor, "American sign language digits," *American Sign Language Digits | Kaggle*, 2023.

[24] M. Xu, S. Yoon, A. Fuentes, and D. S. Park, "A comprehensive survey of image augmentation techniques for deep learning," *Pattern Recognition*, vol. 137, #109347, 2023, doi: 10.1016/j.patcog.2023.109347.

[25] G. S. Nandini, A. P. S. Kumar *et al*., "Dropout technique for image classification based on extreme learning machine," *Global Transitions Proceedings*, vol. 2, no. 1, pp. 111–116, 2021.

[26] S. R. Dubey, S. K. Singh, and B. B. Chaudhuri, "Activation functions in deep learning: A comprehensive survey and benchmark," *Neurocomputing*, vol. 503, pp. 92–108, 2022, doi: 10.1016/j.neucom.2022.06.111.

Philip J. Ezigbo is a faculty member of the School of Electrical Systems Engineering and Technology and a Lecturer/Researcher in the Department of Mechatronics Engineering at the Federal University of Technology Owerri, where he holds an MEng degree with distinction in Control Systems Engineering as well as a BEng degree in Electrical and Electronics Engineering, both of which were awarded by the same institution in 2018 and 2012, respectively. He is currently pursuing a PhD in Control Systems Engineering at the same institution. Engr. Ezigbo's academic

responsibilities include teaching various Engineering courses such as Digital Signal Modeling, Microcontroller & Embedded Systems Design, Mobile Robotics, and Computer Programming for Engineering Applications. His research interests encompass Robotics and Computer Vision, Control & Embedded system, AI and its applications to health and agriculture, IoT, and Cyber-Physical Systems. He is a member of the Council for Regulation of Engineering in Nigeria.

Onyebuchi C. Nosiri received his B.Eng. degree in Electrical and Electronic Engineering, M.Eng. degree in Electronic and Computer Engineering (Telecommunication option) and PhD degree in Telecommunication Engineering from Nnamdi Azikiwe University, Awka Anambra State in 2001, 2009 and 2015 respectively. He is currently an Associate Professor at the Department of Telecommunication Engineering, School of Electrical Systems Engineering and Technology, Federal University of Technology, Owerri, Nigeria and the current Director of the University ICTC. His research interests are in wireless/radio and data communication, Machine learning, Cognitive radio and digital signal processing (DSP). He has more than 12 years of experience in teaching telecommunication engineering courses and has to his credit over 50 journal publications and conference papers. He is a member of some professional bodies such as; the Nigerian Society of Engineering, Council for the Regulation of Engineering and IEEE.

Ekene Samuel Mbonu is a control system expert and R&D consultant who has earned a B.Eng. degree in electronics and computer engineering, an M.Eng. degree in computer engineering, and a PhD degree in computer and control engineering. Currently, he is a lecturer in the Department of Mechatronics FUTO and has previously held positions such as head of the mechatronics department and head of the R&D department at the Electronic Development Institute. Mbonu has experience in automating systems, working with military research teams, and designing embedded and computer systems. He is a member of multiple professional organizations including the Council for the Regulation of Engineering in Nigeria, and has published over 20 technical papers.

Victor Ofor is a final-year undergraduate student of Mechatronics Engineering at the Federal University of Technology Owerri with a focus on Cyber-Physical Systems. He is concurrently pursuing a Statistics and Data Science Micro Master's program at the Massachusetts Institute of Technology, where he is acquiring proficiency in contemporary technologies as well as programming languages for data analysis and machine learning. Victor is driven by a strong inclination to apply his expertise in tackling real-world challenges and developing intelligent systems that can enhance the quality of life and the environment. He is an active member of Hash node, a platform where he shares his insights and knowledge on diverse topics related to mechatronics engineering. He a pivotal role in sourcing and preparing the dataset used in this work and also ensured it was suitable for training purposes.

Jude K. C. Obichere received the BEng and MEng degrees in electrical and electronic engineering (power systems) from University of Port Harcourt, Nigeria, in 1992 and 2004 respectively. He also obtained his PhD in control engineering and renewable energy in 2016 at North Umbria University Newcastle Upon Tyne, United Kingdom where he also worked as an associate lecturer. He worked for so many years in the Oil & Gas Industry on Drilling Mud Solids Control & Waste Management and VSP Seismic. He is currently a Senior Lecturer with the Department of Mechatronics Engineering, Federal University of Technology Owerri (FUTO). He has received Merit Award and Best Paper Award for his outstanding quality research papers, from the World Congress on Engineering and Computer Science (WCECS) of the International Association of Engineers (IAENG). He has co-authored 3 book chapters published in Springer and World Scientific and published over 20 international journals and conference papers. He is a member of the Nigerian Society of Engineers (MNSE), Council for the Regulation of Engineering in Nigeria (COREN), International Association of Engineers (IAENG), The Institution of Engineering and Technology (IET) United Kingdom, European Energy Centre (EEC) and Power & Energy Society Member of IEEE among others.