Natural Human-Computer Interface Based on Gesture Recognition with YOLO to Enhance Virtual Lab Users' Immersive Feeling

Momina Liaqat Ali
Dr. Zhou Zhang, Middle Tennessee State University

I have been an Associate Professor in the Department of Engineering Technology at Middle Tennessee State University since August 2022. Before taking this position, I was an Assistant Professor at the CUNY New York City College of Technology from August 2017 to August 2022. I earned my Ph.D. degree in Mechanical Engineering with the honor of the James Harry Potter Award for outstanding performance in the Doctoral Program at the Stevens Institute of Technology. My research at Stevens is on robotics and virtual reality used in engineering education. My master's degree was in Electrical Engineering, obtained from Southeast University. I received my bachelor's degree in Mechanical Engineering at Southwest Jiaotong University. I have over 7-years of industrial experience as an electrical engineer and mechanical engineer. I also have extensive teaching experience with respect to various interdisciplinary courses involving Mechanical Engineering, Electrical Engineering, and Computer Science.

Natural Human-Computer Interface Based on Gesture Recognition with YOLO to Enhance Virtual Lab Users' Immersive Feeling

Ali, M & Zhang, Z.

Abstract

Hand tracking and gesture recognition are rapidly developing fields with many applications in human-computer interface (HCI). This technology enables computers to recognize and respond to hand movements and gestures, creating a more natural and intuitive interface. With the increasing popularity of augmented reality and virtual reality devices, the demand for advanced hand tracking and gesture recognition technologies is growing. The purpose of this research is to study the current state of the art in hand tracking and gesture recognition and to develop new and improved techniques for HCI applications with 'You only look once' models that result in the improvement of the user's immersive feeling in the virtual world. The research results will be used in a virtual electrical power lab along with the learning management system. To evaluate the implementation, the surveys will be administered before and after the classes. The research will contribute to advancing the technologies by developing new and improved hand tracking and gesture recognition algorithms and integrating them into HCI applications.

Keywords: Virtual Reality, Hand Tracking, Gesture Recognition, YOLO

1. Introduction

1.1 Exploring Frontier of Interaction: Human-Computer Interface in Virtual Reality

Virtual Reality (VR) is a groundbreaking innovation that transcends traditional boundaries between the physical and digital realms [1,2]. The COVID-19 pandemic has accelerated the shift from in-person to remote learning, making VR more significant in education [3,4,5]. A comprehensive VR system consists of five elements: VR engine, software & database, input/output devices, users, and tasks. The system unfolds across three layers: system, middle (input/output devices), and application[6].

As the necessary components of middle layer in VR system, the Human-Computer Interface (HCI) is crucial in VR. They facilitate the interaction between users and digital surroundings. HCI goes beyond hardware and software integration, serving as the conduit for perceiving, manipulating, and communicating within the digital universe. Gestures, gazes, and actions shape immersive experiences, surpassing traditional interfaces such as mouse and keyboard [7]. Therefore, it is so significant to explore the significance of HCI in VR, its role in user experiences, and its influence on virtual environment interactions. This pilot implementation aims to join the journey into the advancements, challenges, and possibilities defining the resilient relationship between humans and the digital realm in VR.

1.2 Challenges in Developing Natural HCI For Virtual Labs

The transition from conventional interfaces to HCI within virtual laboratories presents a host of intricate challenges, underscoring the importance of meticulous attention and inventive problem-solving. These challenges encompass the precise recognition of gestures, real-time responsiveness, accommodation of a wide array of gestures, user-centric design, and smooth integration with

virtual laboratory tasks. Addressing these multifaceted issues necessitates a comprehensive approach that melds advanced Gesture Recognition technologies with user-centric design principles, all while possessing a nuanced understanding of the complexities inherent to virtual lab environments.

Up to now, various head-mounted displays have been introduced in the realm of VR, including the Meta Quest Pro, Apple Vision Pro, HTC Vive Pro, and others [8]. While these devices enhance the user's sense of immersion, their hefty price tags can detract from the natural user experience. Building upon previous implementations that focused on the procedural augmentation of VR, we embark on a journey to devise innovative solutions to these challenges [9]. Our proposed solution revolves around a You Look Only Once (YOLO)-based Gesture Recognition system, potentially serving as a groundbreaking advancement. By navigating the intricacies of precision, responsiveness, adaptability, user-centric design, and seamless integration, our aim is to forge a pathway towards a natural HCI that enriches the immersive and user-friendly aspects of virtual laboratory environments.

2. Gesture Recognition with YOLO

2.1. YOLO Framework

The YOLO framework revolutionizes computer vision and object detection, offering real-time detection through a single-pass architecture [10]. Its unique approach enables swift and accurate object recognition, minimizing computational complexity and supporting applications requiring instantaneous responsiveness (Figure 1). YOLO's characteristics include: (1) Single pass, real-time detection, emphasizing a streamlined process for efficient and instant object detection; (2) Unified framework for object recognition, treating detection as a regression problem, providing a holistic understanding of objects in an image; (3) Efficiency and accuracy, achieved by dividing the input image into a grid for precise, real-time detection without compromising performance; (4) YOLO for gesture recognition, extending its capabilities to enhance the HCI in virtual labs.

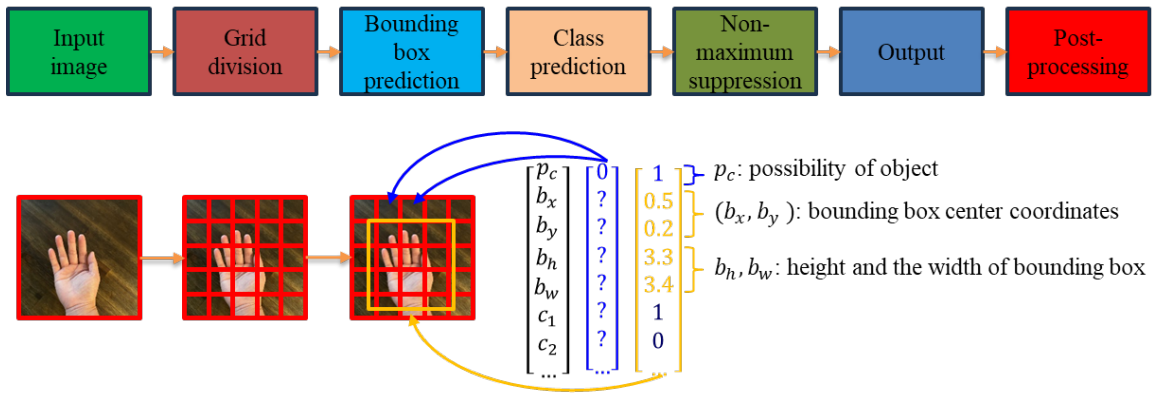

Figure 1: YOLO work flowchart

2.2 Hand Gesture Recognition

To achieve hand gesture recognition, our system employs three key tasks: hand recognition, hand tracking, and hand gesture recognition.

(1) Hand Recognition: Using a webcam, the YOLO framework efficiently identifies and isolates the user's hand in real-time by predicting precise bounding boxes. This initial step lays the foundation for subsequent tasks by establishing a robust identification of the hand's position and boundaries.

(2) Hand Tracking: With the hand recognized, YOLO's grid division and bounding box prediction enable continuous tracking of hand movement across frames. Dynamic updates to the bounding box coordinates ensure smooth and accurate tracking, forming a bridge between initial recognition and subsequent gesture recognition.

(3) Hand Gesture Recognition: Building on tracked hand coordinates, this phase leverages YOLO's class prediction capabilities to associate specific gestures with the tracked hand. Real-time responsiveness ensures seamless recognition as the user performs various hand movements, offering a comprehensive understanding of recognized gestures for natural interaction in virtual environments.

This three-step process, illustrated in Figure 2, underscores the versatility and effectiveness of the YOLO framework in enhancing webcam-based hand gesture interaction.

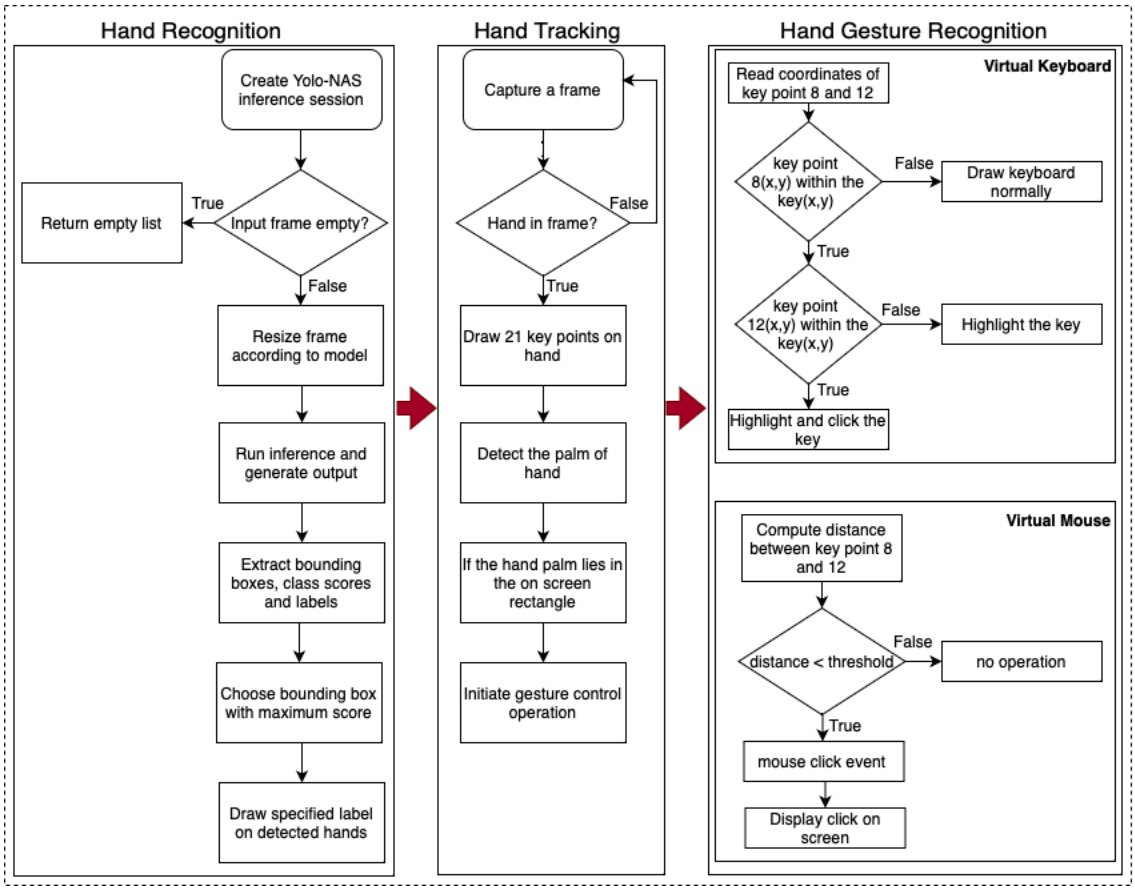

Figure 2: Three-step Process

3. Natural HCI Design Base-on Gesture Recognition

3.1. Gesture Recognition Implementation

This section outlines the methodology employed to achieve hand gesture recognition, tracking, and virtual mouse and keyboard control. The hand posture estimation process consists of two key stages: (i) Data collection and pre-processing, and (ii) Model training and fine-tuning. We curated a diverse dataset featuring various hand orientations, meticulously annotated with 21 landmarks, which underwent thorough pre-processing. Utilizing YOLO-NAS Pose as the chosen object detection model, we successfully identified human hands and pinpointed landmark points. To address the demand for comprehensive data, the model underwent pre-training on a benchmark dataset for posture estimation and fine-tuning using our own images. Ultimately, the detector's optimized weights were leveraged to implement a practical virtual mouse and keyboard application within webcam frames.

3.1.1 Data Collection and Pre-Processing:

During the initial stage, we procured our dataset through the utilization of a webcam, deploying a Python code to capture images at one-second intervals. Given the significance of annotating hand landmarks in our context, we deliberately opted for a green background to enhance the clarity and emphasis on these specific points. The 21 hand landmarks, highlighted in Figure 3, were the focal points of our annotation efforts.

A total of 163 images were amassed, and to augment the dataset's size, we employed a widely used technique called dataset augmentation. This method introduces variations into the dataset through the application of either geometric transformations or kernel filtering operations [11]. Common geometric transformations encompass resizing, flipping, and stretching images, among others, whereas kernel filtering operations involve actions such as blurring and altering the overall resolution of the image. In our specific approach, we opted for kernel filtering over geometric transformations to preserve the integrity of the hand landmarks.

For the data annotation process, we leveraged MMPose to generate annotations in the specified format. MMPose is an integral component of the renowned MMLab framework, an open-source toolkit built on PyTorch [12]. An illustrative example of our dataset, accompanied by its corresponding annotations, is depicted in Figure 3.

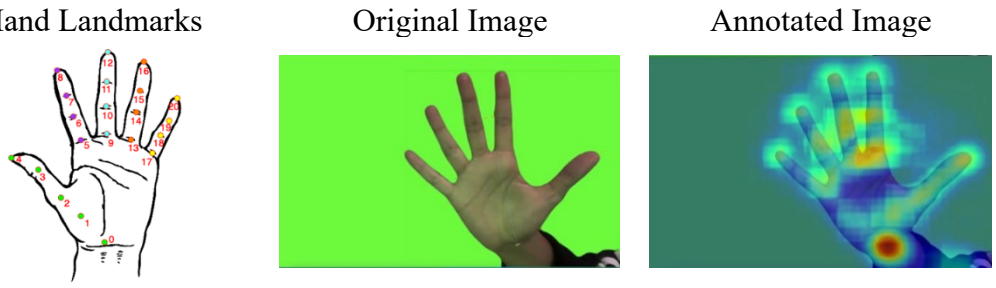

Figure 3: Hand landmarks, instance of customized dataset along with an annotated image.

3.1.2 Model training and fine tuning:

In our study, we opted for a model belonging to the latest additions to the YOLO family, specifically identified as YOLO-NAS Pose, a sibling model of YOLO-NAS. YOLO models have gained popularity for their classification as single-stage detectors, implying their capability to identify all

objects within a single frame simultaneously. This characteristic renders them notably fast and efficient, particularly advantageous for real-life applications.

Typical posture estimation models commonly adopt either top-down or bottom-up approaches [13]. The top-down strategy involves initially identifying objects in the frame and subsequently estimating their poses. However, this approach may falter in crowded frames containing numerous objects of interest. Conversely, the bottom-up approach addresses crowded frames by first pinpointing essential key points and then assembling them into corresponding poses for each individual in the frame. Nevertheless, this approach encounters challenges when faced with occluded objects.

In contrast, YOLO-NAS Pose distinguishes itself by eschewing both top-down and bottom-up strategies. It accomplishes the dual tasks of detecting and estimating poses in a single pass, thanks to its parallel structure implemented in the head module. While the architectures of YOLO-NAS and YOLO-NAS Pose share significant similarities, the key distinction lies in their head architectures. YOLO-NAS's head is solely responsible for identifying objects and their bounding boxes. In contrast, the head architecture of YOLO-NAS Pose is designed not only to identify objects but also to extract pose estimation information from images [14]. A succinct overview of the disparities between the two architectures is presented in Table 1 below.

Table 1: A summarized description of Yolo-Nas and Yolo-NAS Pose

Features	**Yolo-NAS**	**YOLO-NAS Pose**
Objective	Single-Objective: Object Detection and recognition	Multi-Objective: Object detection and recognition along with Posture estimation
Architecture	Backbone, neck and head which contains standard convolutional and FC layers	Same backbone and neck but head is different so to perform multitasking
Loss function	Intersection over Union (IoU)	IoU + key-point similarity score
Key point localization	Not Applicable	Utilizes a regression technique called Object Key-point Similarity (OKS) to perform accurate key-point estimation

The YOLO-NAS Pose model undergoes training on the extensive COCO2017 dataset [15], a vast repository comprising over 200,000 distinct instances of human postures, each meticulously labeled with 17 landmark points. Given the necessity for a substantial dataset in training deep learning models, we employed a fine-tuning approach to enhance the model's performance16. This involved refining the object detector model on our custom dataset, leveraging the knowledge acquired during pre-training on the COCO2017 dataset.

Figure 4 provides a visual representation of the meticulous annotation process, highlighting the 21 essential key points corresponding to the finger joints of a human hand. The fine-tuning process serves as a crucial adaptation mechanism, enabling the model to focus on accurately identifying and tracking hand postures based on the distinctive features within our hand dataset. This not only mitigates the data scarcity challenge and refines the model but also empowers it to deliver precise hand posture estimations, with potential applications in a diverse range of real-world scenarios.

3.2. Natural HCI Design

For the implementation of the virtual mouse and keyboard on a local system, we leveraged the capabilities of OpenCV, a widely recognized Python library renowned for its effectiveness in creating time-responsive computer vision applications[17].

3.2.1 Virtual Mouse:

In our study, we employed the weights derived from our object detection model to accurately identify and track the user's hand. The input for our system was obtained from a standard webcam, typically integrated into laptops, which captured and recorded frames. Subsequently, our program analyzed these recorded frames to determine the hand's position, effectively translating hand movements into cursor positions on the computer screen. To seamlessly replicate mouse and keyboard functionalities, we integrated PyAutoGUI[18], a well-established Python library known for its proficiency in supporting various automation tasks.

Upon detecting a hand in the frame, the algorithm utilizes the mean key-point score to determine the centroid of the palm. If the centroid aligns with the pre-drawn rectangular region on the screen, the gesture control function is activated. Subsequently, using PyAutoGUI, the mouse pointer is repositioned to the center of the screen. The cursor can then be manipulated by moving the hand. If no hand is detected in the camera view, the gesture control operation is disabled. Key-point 8, corresponding to the tip of the index finger, plays a crucial role in virtually rotating the mouse, while key-point 12 represents the tip of the middle finger. These key-point IDs are illustrated in Figure 3. When the distance between key-point 8 and 12 is significantly small, a click operation is triggered. The comprehensive workflow of the virtual mouse is depicted in Figure 4.

3.2.2 Virtual Keyboard:

We developed a virtual keyboard featuring an intuitive user interface displayed on the screen. Upon launching the application, an empty list of characters is initialized to record user input. The tracking system, focused on key-points 8 and 12 representing the middle and index fingertips, respectively, is employed to detect button presses on the virtual keyboard. Figure 5 provides a snapshot of our designed virtual keyboard.

The algorithm for detecting key presses in our virtual keyboard is succinctly outlined below:

(1) Determine the size and position of keys on the on-screen keyboard based on the keyboard layout.

(2) Check if the fingertips of the index and middle fingers fall within the boundary of a key.

(3) Highlight the key when both the middle and index fingertips have their tips within the key boundary, updating the on-screen text accordingly. If the key is 'BK', remove a character from the list of input characters.

(4) If only the index finger is on the key, highlight that key.

(5) If neither fingertip is on a key, display the keyboard normally.

In Figure 6, the detailed flow of our virtual keyboard interface is depicted. Initially, the coordinates of the index finger are computed, and if they fall within the boundaries of any key on the on-screen keyboard, the coordinates of the middle fingertip are then checked. If both the x-axis and y-axis values of the fingertips and keyboard buttons fall within the same range, visual feedback is

provided as the corresponding key on the virtual keyboard becomes highlighted and is considered pressed. Concurrently, the list of input keys is dynamically updated as new keys are pressed. This interactive design facilitates easy character input by allowing users to type with the natural motion of their fingers.

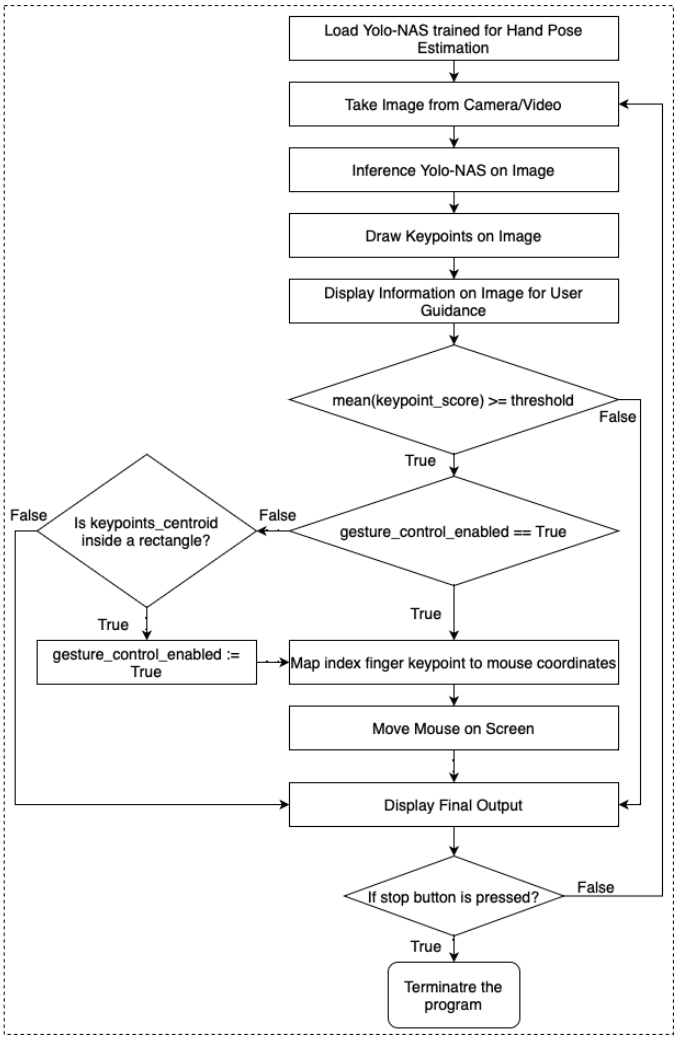

Figure 4: Detailed flowchart of virtual computer mouse.

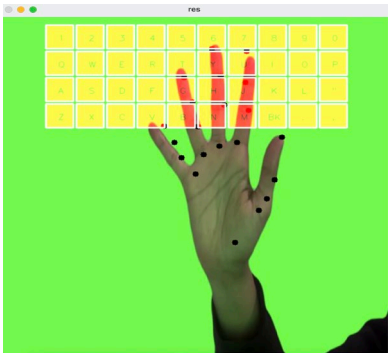

Figure 5: On-screen virtual keyboard application.

4. Application of Natural HCI in Virtual Electrical Power Lab

4.1. Virtual Electrical Power Lab: LVSIM-EMS

LVSIM-EMS, the Electromechanical Systems Simulation Software, encompasses various FESTO-developed systems, including the Computer-Assisted 0.2 kW Electromechanical Training System, DC and AC Power Circuits Training System, Electromechanical Training System, and AC Power Transmission Training System. Accessible in the navigation menu, LVSIM-EMS provides online reference workbooks for these systems, replacing traditional EMS laboratory equipment with virtual modules for manipulation on computer screens. The software employs sophisticated mathematical models to simulate the electrical and mechanical characteristics of real EMS

modules, offering cost-effective, versatile exercises on electrical power and machines. Available for local installation on Windows PCs, servers, or online, LVSIM-EMS facilitates learning in topics such as active and reactive power, phasors, AC/DC motors and generators, three-phase circuits, and transformers [19].

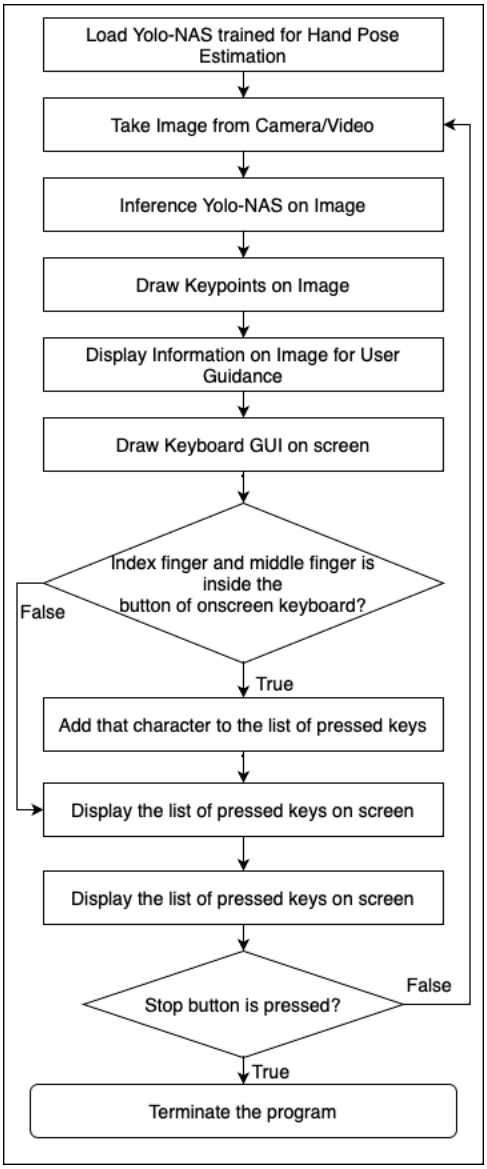

Figure 6: Detailed flowchart of virtual keyboard.

4.2. Validation and Evaluation by Labs in Mechatronics and Electrical Engineering Courses

4.2.1 Description of courses experiments

In the context of the Electrical Engineering curriculum at MTSU, two specific courses, namely ENGR 4520 Electrical Power and Machinery, and ET 4640 Industrial Electricity, traditionally

conducted experiments utilizing power trainers exclusively [20]. However, with the introduction of the LVSIM-EMS platform, there has been a notable shift in the experimental methodology.

The integration of LVSIM-EMS has brought about a transformative approach to these courses. Students are now tasked with proving and validating experimental methods using the virtual simulation environment provided by LVSIM-EMS before physically engaging with the traditional power trainers. This shift in approach offers students a unique opportunity to bridge theoretical knowledge with hands-on practical application.

The curriculum experiments in both ENGR 4520 and ET 4640 leverage the capabilities of LVSIM-EMS to simulate various electrical power and machinery scenarios. Students engage in a step-by-step process of planning, executing, and validating experiments within the virtual environment, allowing them to gain a deeper understanding of the underlying principles. Following the virtual simulation phase, students transition to the physical power trainers, where they can implement and verify the same experimental methods learned in the virtual setting. This dual-phase approach not only enhances the students' comprehension of electrical power and machinery concepts but also reinforces the connection between theoretical knowledge and real-world applications.

In summary, the incorporation of LVSIM-EMS in ENGR 4520 and ET 4640 courses at MTSU has transformed the traditional experimental methods. By requiring students to first demonstrate their understanding in a virtual environment, this approach provides a comprehensive learning experience that extends beyond theoretical concepts to practical, hands-on applications using traditional power trainers.

4.2.2 Validation and evaluation

In the original HCI setup for the LVSIM-EMS laboratory, students primarily utilized traditional input devices such as a mouse and keyboard to connect and operate various devices within the simulation environment. However, a significant advancement has been made with the introduction of the natural HCI approach, offering students an immersive and hands-on experience.

The natural HCI paradigm allows students to mimic realistic wiring procedures when completing circuit connections and operating devices within the LVSIM-EMS software. Unlike the conventional mouse and keyboard interface, the natural HCI approach integrates more intuitive and life-like interactions, closely resembling the actual physical processes involved in handling electrical circuits.

Taking the Short Circuit Test of a single-phase transformer, as depicted in Figure 7, as an illustrative example, students now have the opportunity to engage in a more authentic and practical experience. Instead of relying solely on traditional input methods, they can virtually replicate the steps involved in a real-world wiring procedure. This not only enhances the learning process but also provides a seamless transition from theoretical understanding to practical application.

The integration of natural HCI in the LVSIM-EMS lab seamlessly aligns with the ongoing pedagogical shift towards experiential learning. This innovative approach ensures that students are immersed in a dynamic and engaging educational environment, providing them with the opportunity to develop skills closely aligned with those demanded in the professional field. The natural HCI method elevates the educational experience by fostering a profound understanding of electrical concepts through a hands-on and interactive interface. This not only enriches learning but also serves as a crucial preparation for students to navigate real-world scenarios in the field of electrical engineering. Moreover, the incorporation of the virtual lab component serves to minimize the risk

of damage to the power trainer caused by short circuits, contributing to a safer and more controlled learning environment.

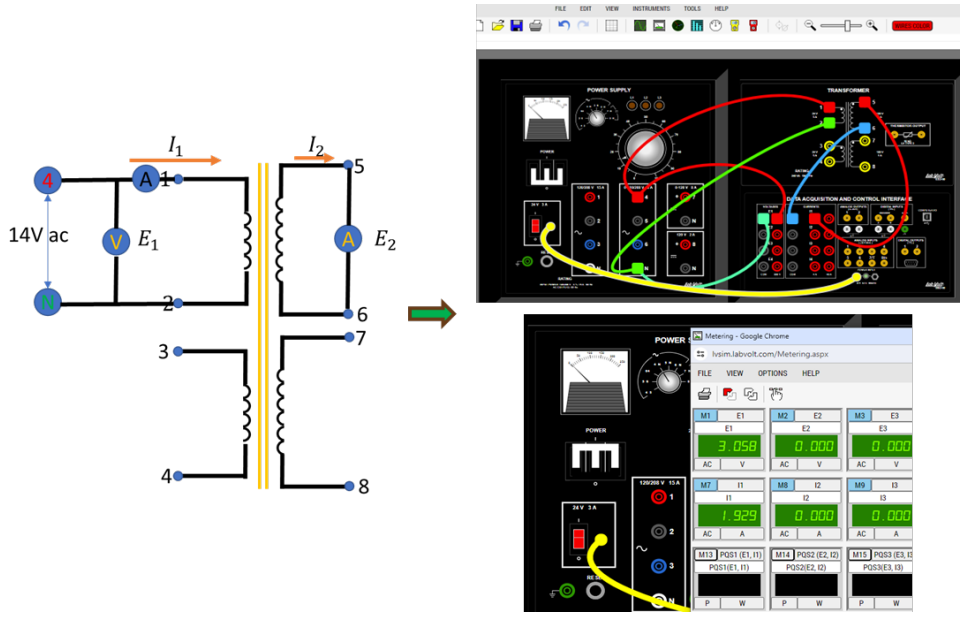

Figure 7: Instance of the application: Short Circuit Test of a single-phase transformer.

In each semester, there are two classes, totaling 106 students. Half of the students engaged directly with the software utilizing traditional HCI), while the remaining half employed natural HCI. The evaluation encompassed a comprehensive comparison of students' performance across key metrics: (i) the ability to integrate knowledge and information for problem identification; (ii) demonstrated understanding of all facets of a problem; (iii) formulation of strategies to solve narrowly defined problems; (iv) identification of correct and detailed solutions to problems; and (v) problem-solving proficiency in both midterm and final assessments. Detailed evaluation results outlining students' performance can be found in Figure 8.

5. Conclusions and Future Work

In summary, our research focuses on elevating HCI within VR environments, specifically delving into hand tracking and gesture recognition technologies. Utilizing YOLO models, we showcase the feasibility of natural HCI in a virtual electrical power lab, enhancing user interaction in VR settings. The integration of YOLO frameworks and the development of innovative virtual mouse and keyboard interfaces represent significant strides in improving the immersive quality of VR experiences. Moreover, the application of our research in mechatronics and electrical engineering courses at MTSU, utilizing LVSIM-EMS, signifies a transformative approach to hands-on learning by bridging theory and practical application.

Looking ahead, future work involves continual refinement of hand tracking algorithms for increased accuracy and recognizing a broader range of gestures. Exploring advanced algorithms and wearable sensor haptic feedback devices can enhance the precision and realism of virtual interactions. Longitudinal studies assessing the long-term impact of natural HCI in education and further

investigations into the incorporation of emerging technologies will contribute to the continuous advancement of the field.

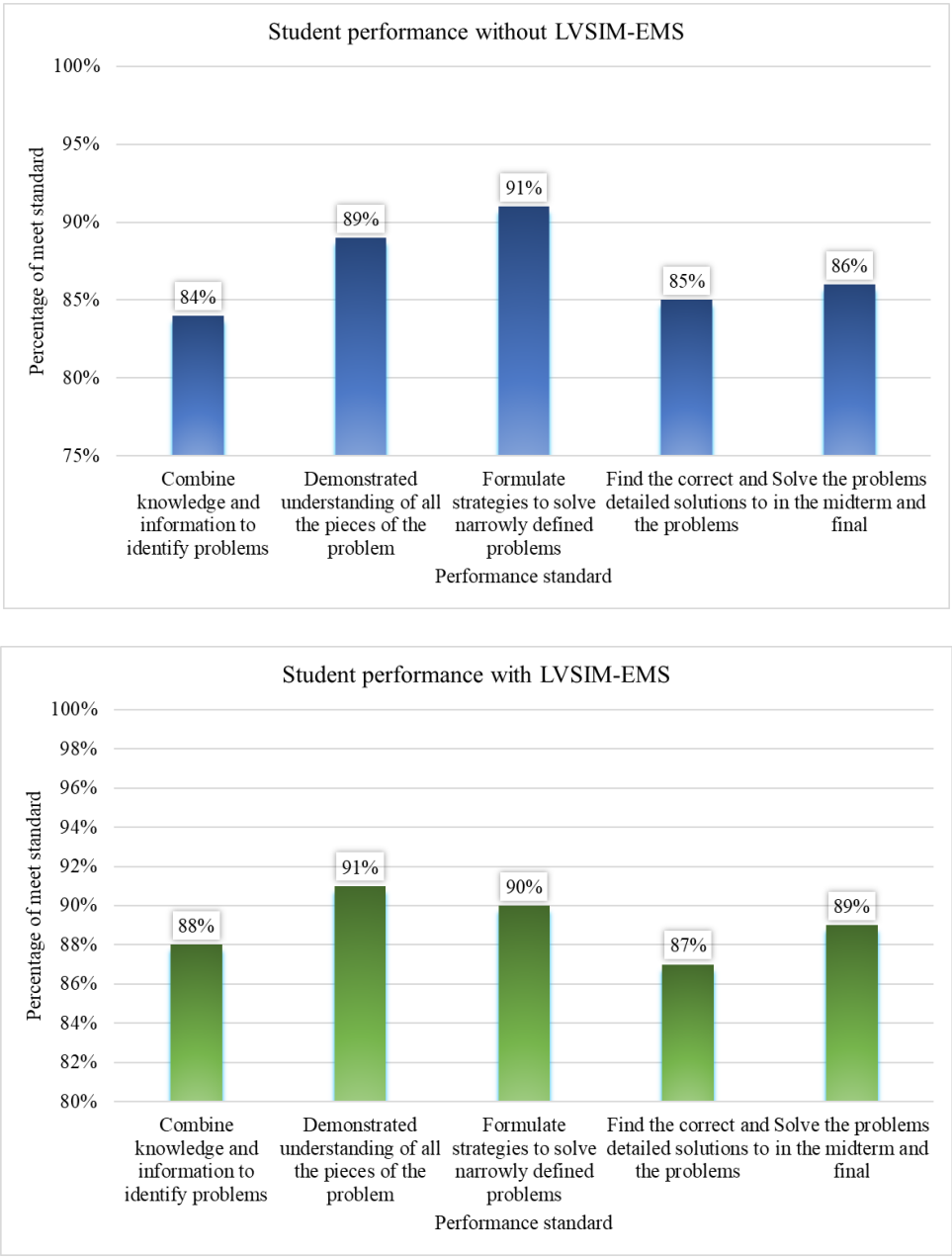

Figure 8: Statistic of students performance.

References

[1] Brey, P., 2014, "Virtual reality and computer simulation", In: Ethics and Emerging Technologies, pp. 315-332, Palgrave Macmillan UK.

[2] Kozak, I., Banerjee, P., Luo, J. & Luciano, C., 2014, “Virtual reality simulator for vitreoretinal surgery using integrated OCT data”, Clinical Ophthalmology, Vol. 8, pp. 669-672.

[3] Zhang, Z., Chang, Y., Esche, S.K. and Zhang, A.S, 2022, “Application of internet of things in online robotics class”, ASEE Annual Conference & Exposition, Minneapolis, Minnesota, USA, July 26-29, 2022.

[4] Alfaisal, R., Hashim, H. & Azizan, U.H.,2024, “Metaverse system adoption in education: a systematic literature review”, J. Comput. Educ, Vol. 11, pp. 259-303.

[5] Jeyakumar, T., Ambata-Villanueva, S., McClure, S., Henderson, C., & Wiljer, D., 2021, “Best practices for the implementation and sustainment of virtual health information system training: qualitative study”, JMIR Med Educ, Vol. 7, No. 4.

[6] Zhang, Z., Zhang, A.S., Zhang, M. & Esche, S.K., 2018, “Immersive educational systems with procedure-oriented combinations of real and virtual environments”, Proceeding of ASME International Mechanical Engineering Conference & Exposition IMECE’18, Pittsburgh, PA, USA. November 9-15, 2018.

[7] Halabi, N., Jones, E. & Mirza-Babaei, P., 2023, “Unleashing immersive experiences: the power of gesture-based VR interaction”, Interactions, Vol. 30, No. 6, pp. 52-55.

[8] https://www.nytimes.com/wirecutter/reviews/best-standalone-vr-headset/, accessed in March, 2024.

[9] Zhang, Z., Zhang, A.S., Zhang, M. and Esche, S.K., 2018, “Immersive educational systems with procedure-oriented combinations of real and virtual environments”, Proceeding of ASME International Mechanical Engineering Conference & Exposition IMECE’18, Pittsburgh, PA, USA. November 9-15, 2018.

[10] Redmon, J., Divvala, S., Girshick, R. & Farhadi, A., 2016, “You only look once: Unified, real-time object detection”, Proceedings of the IEEE conference on computer vision and pattern recognition Las Vegas, NV, USA, June 27-30, 2016.

[11] Shorten, C. & Khoshgoftaar, T. M., 2019, "A survey on image data augmentation for deep learning" Journal of Big Data, Vol. 6, No. 60, pp. 1-48.

[12] https://mmpose.readthedocs.io/en/latest/advanced_guides/customize_datasets.html, “Dataset Annotation and Format Conversion — MMPose 1.3.1 Documentation, n.d.”, accessed in March, 2024.

[13] Kuo, P., Makris, D., &Nebel, J.-C., 2011, “Integration of bottom-up/top-down approaches for 2D pose estimation using probabilistic Gaussian modelling”, Computer Vision and Image Understanding, Vol. 115, pp 242–255.

[14] GitHub - Open-Mmlab/Mmyolo: OpenMMLab YOLO Series Toolbox and Benchmark. Implemented RTMDet, RTMDet-Rotated,YOLOv5, YOLOv6, YOLOv7, YOLOv8,YOLOX, PPYOLOE, Etc., n.d.; Terven et al., 2023

[15] https://docs.ultralytics.com/datasets/pose/coco/, accessed in March, 2024.

[16] Anaby-Tavor, A., Carmeli, B., Goldbraich, E., Kantor, A., Kour, G., Shlomov, S., Tepper, N., & Zwerdling, N., 2020, “Do not have enough data? deep learning to the rescue!”, Proceedings of the AAAI Conference on Artificial Intelligence, New York, USA, February 7-12, 2020.

[17] Dardagan, N., Brdanin, A., Dzigal, D., & Akagic, A., 2021, “Multiple object trackers in OpenCV: a benchmark”, Proceeding of International Symposium on Industrial Electronics, June 20-23, 2021.

[18] https://pyautogui.readthedocs.io/en/latest/, accessed in March, 2024.

[19] https://labvolt.festo.com/solutions/6_power_energy/98-8970-00_electromechanical_systems_simulation_software_lvsim_ems, accessed in March, 2024.

[20] https://labvolt.festo.com/solutions/6_power_energy/59-8010-90_electromechanical_training_system,accessed in March, 2024.