



Visionary Hand Rehab: Enhancing Grasp Assistance using Artificial Intelligence with Functional Electrical Stimulation

Abhay Kshirsagar

Aditi Mishra

Dept. of Electronics Engineering

*Vivekanand Education Society's Institute of Technology
Mumbai, India*

Ameya Roplekar

Mehul Nikumbh

Dept. of Electronics Engineering

*Vivekanand Education Society's Institute of Technology
Mumbai, India*

Vedant Jadhav

Dept. of Electronics Engineering

*Vivekanand Education Society's Institute of Technology
Mumbai, India*

Abstract—Stroke ranks as the second leading cause of death in India, with approximately 1,85,000 cases occurring annually. A common consequence of stroke is hand paralysis, necessitating effective rehabilitation strategies such as Functional Electrical Stimulation (FES) for hand prehension. Existing commercial FES devices often pose usability challenges due to manual operation via touchpad or buttons. In this study, we propose a novel deep learning-based approach using the YOLOv8 algorithm for object detection to facilitate multiple grasp types and control an electrical stimulator for assisting grasping. Our algorithm tracks hand and object positions, selecting appropriate grasp types based on proximity to the hand. Upon selection, an ESP32 BLE device, which imitates specific behaviors of a custom FES stimulator, executed predefined stimulation sequences to enable the user to grasp nearby objects. This contactless, vision-based solution holds promise for home-based stroke rehabilitation by eliminating setup complexities and reducing dependence on caregivers. Future research will integrate this model with FES devices, evaluating its efficacy in restoring hand movements among stroked patients.

Index Terms—object detection, object tracking, yolov8, ble device, esp32, functional electrical stimulation, grasp mechanism, web application, human-machine interface

I. INTRODUCTION

Worldwide, stroke is the commonest cause of mortality after coronary artery disease. Also, it is the commonest cause

of chronic adult disability. The lifetime risk of stroke after 55 years of age is 1 in 5 for women and 1 in 6 for men [1]. According to incomplete statistics, nearly 80% of stroke patients develop hand dysfunction, one of the most common symptoms of stroke. Hand function accounts for about 90% of upper limb function and plays an indispensable role in life. Hand dysfunction seriously affects the quality of daily life of patients and their families, and has a negative impact on patients' psychology, so hand function rehabilitation has become one of the main tasks of stroke rehabilitation [2].

One of the most promising approaches to improve motor function in complete and incomplete paraplegic and tetraplegic patients with permanent limb impairment is the Functional Electrical Stimulation (FES).

FES is a methodology that uses bursts of short electrical pulses to generate muscle contraction. If these electrical pulses are applied to motor nerves, they can elicit Action Potentials (AP) that propagate along the axons towards the target muscle. Once these AP reach the muscle, they cause muscle contraction. To achieve a continuous muscle contraction (tetanization) the FES system must induce at least 20 AP per second, otherwise the muscle would not generate a steady output force and would only twitch. In principle, the motor nerves can be stimulated using monophasic and biphasic current or voltage

pulses. Since monophasic pulses can potentially cause skin burns and tissue damage due to the galvanic processes, most FES systems implement biphasic current pulses [3].

Using FES, we can safely and effectively restore grasping, shoulder and elbow movements, sit-to-stand ability and walking in paralyzed individuals [4]. However, the current generation of commercial Functional Electrical Stimulation (FES) devices designed for grasping presents significant challenges in usability for patients. These challenges primarily stem from the manual operation required, typically involving touchpad or button controls. This manual interface not only complicates usage for patients but also has the potential to reduce their engagement during therapy sessions. Therefore, there is an urgent need to develop automatic control systems for FES devices that integrate with Human-Machine Interfaces (HMI). This integration aims to simplify operation and enhance patient involvement in therapy sessions.

The shape of an object often determines the type of grasping our hands perform to pick up that object. For example, a cylindrical grasp is suitable for grasping curved surfaced objects such as a cup or hammer, while the pinch grasp is suitable for tasks requiring fine motor skills such as picking up a needle or pen. Hence, by identifying the object that a person is intending to grasp (based on its shape), we can infer the grasp type that must be employed to pick up that object. One approach to determining object type is to use camera-based object detection [5]. Utilizing object detection will enable the simultaneous monitoring of multiple objects positioned within the person's reach. By tracking the proximity of these objects to the person's hand, it will become feasible to determine the specific object intended for grasping. Different research has been done in recent years, but because of different challenges such as occlusion, illumination variations, fast motion, etc. research in this area continues [6]. This capability can be leveraged to facilitate the grasping process through a Functional Electrical Stimulation (FES) device, enhancing the precision and effectiveness of assisted grasping interventions.

A. Related Work

Previous research has utilized object detection to determine grasp types and control Functional Electrical Stimulation (FES) devices [7]. However, the study employed a single-stage detector, specifically YOLOv5 (You Only Look Once, version 5), which utilizes a single CNN output for detecting object locations (via bounding boxes) and classifying object types (with associated probabilities). Additionally, integration of the final model into a web application for real-time monitoring was not addressed in the study. To advance this research, we propose implementing an improved algorithm and implementing object tracking. The human hand has a complex and fully articulated anatomical structure suitable for performing coarse and fine-grained movements. The development of real-time, robust, non-invasive, cost-effective, and accurate algorithms for tracking human hand and finger movements is complex. We

used Google MediaPipe Hand (GMH) and GMH-D to solve this problem. The latter is an enhanced version of GMH that runs on top of an RGB-D camera, which provides simultaneous and calibrated color and depth video streams [8]. In the end we would integrate the model into a dedicated website. This platform will enable users to connect their devices, facilitating real-time monitoring of their progress and enhancing user engagement with the system.

B. Background

Object detection is a crucial task in computer vision that has its application in various fields like robotics, medical imaging, surveillance systems, and autonomous vehicles. The newest version of the YOLO model, YOLOv8 is an advanced real-time object detection framework, which has attracted the attention of the research community. Of all the popular object identification methods and machine-learning models such as Faster RCNN, SSD, and RetinaNet, YOLO is the most popularly known method in terms of accuracy, speed, and efficiency. This research study provides an analysis of YOLO v8 by highlighting its innovative features, improvements, applicability in different environments, and a detailed comparison of its performance metrics to other versions and models [9].

The YOLOv8 model offers five different variants in terms of size: nano (n), small (s), medium (m), large (l), and extra-large (x). These variants vary in complexity, inference speed, and accuracy levels. In our study, we conducted an evaluation of the performance of these different YOLOv8 models using our custom dataset. This evaluation aimed to assess their accuracy across various tasks and compare their effectiveness in meeting specific performance metrics.

Instead of utilizing a custom FES device for connectivity with the model, we opted for the ESP32, a Bluetooth Low Energy (BLE) device. The ESP32 microcontroller stands out for its dual-core processing capability and built-in Wi-Fi and Bluetooth connectivity, enabling seamless wireless communication and control. This feature is essential for integrating with external sensors or user interfaces. Like a custom FES device designed for delivering therapeutic electrical impulses to targeted muscles, the ESP32 triggers specific actions, such as blinking an LED, as part of our project. This decision was made to reduce project costs significantly. The integration process with a custom FES device would mirror the approach we employed with the ESP32, ensuring consistency in functionality and operation.

II. METHODS

In subsequent sections, we will construct a deep learning-based object detection system tailored for ESP32 control, showcasing its application in aiding with grasping tasks.

This methodology provides a robust and efficient solution by leveraging advanced real-time object detection and tracking technologies, such as YOLOv8 and Google MediaPipe, to ensure precise and reliable object identification and hand

TABLE I
CORRELATION BETWEEN LEDS AND OBJECTS

LED	Correlation
LED 1 (Red)	Ball
LED 2 (Red)	Credit Card
LED 3 (Red)	Marker
LED 4 (Red)	Cup
LED 5 (Green)	Successful Grip

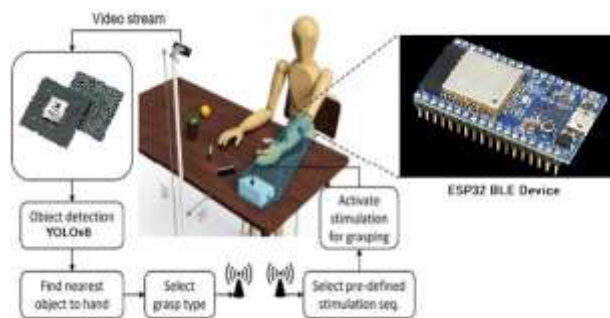


Fig. 1. Schematic diagram for camera-based object detection and grasp type selection. Grasp selection is wirelessly communicated to the BLE device, which then executes programmed stimulation sequences by blinking associated LED.

movement tracking. The integration of these technologies into a user-friendly system demonstrates the practicality and effectiveness of the approach, making it an innovative application and potential impact in the field of rehabilitation technology.

A. System Design and Experiment Setup

Real-time object detection was executed utilizing the NVIDIA MX330 GPU. Additionally, solutions for issues related to detection, tracking, and object movement and appearance were also implemented.

As depicted in Fig. 1, a full HD 1080p webcam (1920x1080 pixels, CASE U) was linked to a laptop and securely mounted on a tripod positioned 70 inches above the ground and 30 inches away from a table. The camera was angled to capture a comprehensive top-down view of the entire tabletop surface, where three common objects - a ball, a credit card, and a marker were placed.

Communication between the laptop and ESP32 occurs via Bluetooth. Patients initiate this connection through a website hosted on the laptop. They select objects they intend to attempt gripping from those displayed. Subsequently, the camera identifies the nearest object to the patient's hand and transmits a corresponding signal to the ESP32. Confirmation of signal reception prompts the ESP32 to activate an associated LED.

Patients then attempt to grasp the selected object. A successful grip triggers a "GRIP" message and causes the ESP32's associated LED to blink. Conversely, an unsuccessful attempt prompts a "NO GRIP" message. This process effectively verifies that the FES device worn by the patient successfully transmits electrical signals to stimulate the patient's fingers, thereby validating the concept and its functionality.

The table below illustrates the correlation between LED pins and detected object:

B. Object Detection using YOLOv8

The default YOLOv8 models (n, s, m, 1, x) are pre-trained on the Microsoft Common Objects in Context (COCO) dataset, encompassing 80 object categories. However, as the COCO

dataset does not cover all the object classes utilized in this study; we retrained the YOLOv8 models using a custom dataset. This dataset comprised images of four objects: a ball, a credit card, a marker, and a human hand. Approximately 70% of the images contained single objects, 10% depicted two or three objects side-by-side (e.g., a cup and a marker), and the remaining 20% featured an experimenter grasping each object. A total of 6347 images, containing 9472 instances of the five object classes, were captured and divided into 5947 training images and 298 validation images. Annotation and labeling of the objects were performed using Roboflow Annotate (Roboflow Inc.), and the data was exported in YOLOv8's annotation format. The models were trained and validated on Google Colab using a Tesla T4 GPU, with training conducted over 50 epochs, a batch size of 16, and an image size of 640 x 640, employing default hyperparameters.



Fig. 2. Detection of ball

Fig 2 illustrates the detection of the ball, with the green box indicating the object's boundary and the associated class ID, i.e. 0. Similarly, Fig3 depicts the detection of the credit card, which is associated with class ID 2, Fig 4 demonstrates the detection of the marker, associated with class ID 4 and Fig 5 demonstrate the detection of the cup, associated with class ID 5.

C. Object Tracking using Google MediaPipe

Although object detection can identify and count similar objects based on shared labels, the development of an efficient and accurate object tracking algorithm is crucial for this study. MediaPipe addresses these challenges by abstracting and connecting individual perception models into maintainable pipelines [10]. The primary objective is to devise



Fig. 3. Detection of credit card



Fig. 6. Mediapipe Hand Landmarker model



Fig. 4. Detection of marker



Fig. 5. Detection of cup

D. Using Object Tracking to send signal to the ESP32

To ensure that when a particular object is detected, appropriate signal is given to BLE device which is a representation of the grasp type to be selected in the hand rehabilitation device. We developed a seamless communication protocol to monitor this action and integrated the code within the web application.

E. Deployment of the Web Application

To support our research, we designed an interactive interface that provides real-time feedback on object detection, hand tracking, and grasp selection. The back-end functionality was implemented to manage user sessions, store rehabilitation data, and facilitate communication between the web interface and the deployed object detection system. We integrated the object detection and hand tracking functionalities with the web application, ensuring seamless interaction between the front end and back-end components. Additionally, the web application was rigorously tested for usability, performance, and reliability. User feedback was solicited, and necessary refinements were made to improve the overall user experience. The deployment of the web application follows the following algorithm:

- 1) Initialization: Import required libraries, including OpenCV, Flask, SocketIO, and others, and initialize the Flask app and SocketIO.
- 2) Define CamShift Class: Define a class named CamShift for object tracking using the CamShift algorithm, initialize class variables such as tracking flag, region of interest, and others, and define methods to set YOLO coordinates, calculate hue, set tracker coordinates, and perform tracking.
- 3) Global Variables and Parameters: Declare global variables for selected classes, YOLO model, hand detection model, and others, and set parameters like frame skipping factor, grip duration threshold, etc.
- 4) Generate Video Frames Function: Define a function to generate video frames with object detection and

tracking, looping while on the video page to read frames from the camera, perform hand detection using MediaPipe, perform object detection using YOLOv8, extract bounding boxes, classes, names, and confidences, update the UI and send SocketIO messages based on detected objects, perform object tracking using CamShift if conditions are met, and encode frames as JPEG and yield them.

- 5) Flask Routes: Define routes for the Flask app, where the index route renders the initial HTML template, the submit route handles form submission and updates selected classes, the results route renders the results HTML template, the video_feed route streams video frames, and additional routes retrieve the count variable and button state.
- 6) SocketIO Connection Event: Define a SocketIO connection event to handle the connection when a BLE device connects.
- 7) Start Flask App: Start the Flask app with SocketIO support.



Fig. 7. Dashboard for connecting BLE devices



Fig. 8. Dashboard for selecting objects

III. RESULTS

A. Training and Validation of Object Models on Custom Dataset

By researching several experiment results such as detection results, distance traveled experiment results, confusion matrix, and experiment results on validation dataset, YOLOv8 proved to be the best for our needs [11][12].

We trained and validated four variants of the YOLOv8 network (s, m, l, x) using our custom dataset. To evaluate the performance of these models, we utilized the Precision-Recall (P-R) curve to calculate their Mean Average Precision (mAP). As presented in Table II, mAP was determined at an intersection-over-union (IoU) threshold of 50% overlap between predicted and actual object bounding boxes (mAP@0.5). Table II also details each model's inference time on a Tesla T4 GPU and the number of parameters trained for each model. The results indicate that all models demonstrated comparable effectiveness in detecting various object classes. Notably, the YOLOv8n model exhibited the shortest inference time and required fewer parameters to be trained, leading to a significant reduction in training time. Consequently, the YOLOv8n model was selected for real-time object detection and deployed on the NVIDIA MX330 GPU.

TABLE II
PERFORMANCE OF DIFFERENT OBJECT DETECTION MODELS ON CUSTOM DATASET

Model Type	mAP@0.5	Inference time (ms)	Parameters
YOLOv8n	92.9%	6.9	7.1 M
YOLOv8s	92.2%	11.7	7.1 M
YOLOv8m	93.4%	20.3	19.9 M
YOLOv8l	94.1%	32.4	49.5 M
YOLOv8x	94.3%	55.5	88.2 M

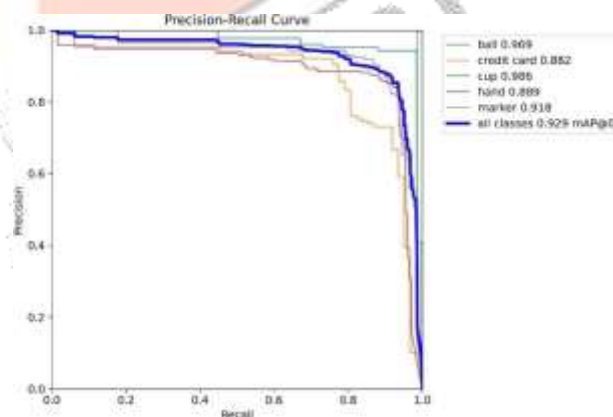


Fig. 9. Precision-Recall (P-R) curve for YOLOv8n neural network on the validation dataset

From the precision-recall curves shown in Fig. 9, we see that the custom YOLOv8n model's precision is lower for hand detection. This is likely because our training sample's size isn't large enough to account for the different hand poses and orientations of a person's hand.

Thus, we used Google's open-source MediaPipe Hands machine learning pipeline that was trained using 30K samples of real-world hand images. We integrated our object detection model with the MediaPipe framework to improve its precision for hand detection.

B. Real-time Control of ESP32

Fig. 10 demonstrates an example of real time control of ESP32 using object detection and grip success demonstration on the website when demonstrated by an able-bodied volunteer. During a test run, our system predicted the user's intentions and assisted grasping with 90% accuracy.



Fig. 10. Displaying the "GRIP" message on successful grip

C. Enhancements/Improvements made

The previous research and experiments reported a mean Average Precision (mAP) of 93.8% at an Intersection over Union (IoU) threshold of 0.5, with an inference time of 8.1 ms and a model size of 7.0 million parameters [7]. As presented in Table II and illustrated in Fig. 9, our current work achieves a mAP of 92.9% at the same IoU threshold, with an improved inference time of 6.9 ms and a slightly increased model size of 7.1 million parameters.

Additionally, we have enhanced the accuracy of hand detection by integrating Google's MediaPipe, which has resulted in an accuracy increase to approximately 99%. This integration, along with embedding the model into a remote website, has significantly improved the system's accessibility and user-friendliness.

IV. DISCUSSIONS

This paper presents a novel approach to grasp assistance and hand rehabilitation that leverages advancements in artificial intelligence (AI) and electrical stimulation technology. The study integrates state-of-the-art AI algorithms with an ESP32 microcontroller configured to mimic Functional Electrical Stimulation (FES). This interdisciplinary collaboration demonstrates significant potential for innovation to revolutionize healthcare and improve patient outcomes.

The proposed system, termed "Visionary Hand Rehab," offers a promising solution for individuals with hand impairments. It paves the way for a future where personalized treatment plans and AI-driven rehabilitation become commonplace. This study not only expands the boundaries of rehabilitation but also lays the groundwork for future research aimed at enhancing patient quality of life globally.

The literature highlights various FES control methods for grasping that utilize wearable sensors to monitor biopotentials [13] or track arm trajectories [14]. However, the vision-based, contactless approach presented here offers distinct advantages for both clinical and home-based rehabilitation settings. It eliminates the need for additional sensors on the patient, improving comfort and reducing setup time. Furthermore, this approach can be seamlessly integrated into telerehabilitation, allowing therapists to remotely monitor therapy sessions by receiving webcam feeds through a cloud-based server [15].

Additionally, implementing a success rate tracking system with the help of data analytics that collects data on the user's grasp attempts during rehabilitation sessions can provide therapists with quantitative assessments, including the number of objects grasped, movement quality metrics, and total therapy duration [16].

ACKNOWLEDGMENT

The authors would like to thank Dr. Nikunj Bhagat, Mr. Sagar Bhokare, Dr. Abhay Kshirsagar, Neuralaxy LLC, Vivekanand Education Society's Institute of Technology, Mumbai for their constant guidance and support in conducting this research.

REFERENCES

- [1] S Seshadri, A Beiser, M Kelly-Hayes, CS Kase, R Au, WB Kannel, PA Wolf, "The lifetime risk of stroke: estimates from the Framingham Study," *Stroke*, vol. 37, no. 2, pp. 345-350, Jan. 2006, doi: 10.1161/01.STR.0000199613.38911.b2
- [2] C Liu, J Lu, H Yang, K Guo, "Current State of Robotics in Hand Rehabilitation after Stroke: A Systematic Review," *Applied Sciences*, vol. 12, no. 9, pp. 4540, Apr. 2022, doi: 10.3390/app12094540
- [3] MR Popovic, A Curt, T Keller, V Dietz, "Functional electrical stimulation for grasping and walking: indications and limitations," *Spinal cord*, vol. 39, no. 8, pp.403-412, Aug. 2001, doi: 10.1038/sj.sc.3101191
- [4] KT Ragnarsson, "Functional electrical stimulation after spinal cord injury: current use, therapeutic effects and future directions," *Spinal cord*, vol. 46, no. 4, pp. 255-274, Apr. 2008, doi: 10.1038/sj.sc.3102091
- [5] M Tiwari, R Singhai, "A review of detection and tracking of object from image and video sequences," *International Journal of Computational Intelligence Research*, pp. 745-765, Mar. 2017
- [6] Z. Soleimanitaleb, M. A. Keyvanrad and A. Jafari, "Object Tracking Methods:A Review,"*2019 9th International Conference on Computer and Knowledge Engineering (ICCCKE)*, pp. 282-288, Oct. 2019, doi: 10.1109/ICCCKE48569.2019.8964761
- [7] NA Bhagat, M Ruppia, "Using Object Detection to Select Grasp Type & Control Functional Electrical Stimulation for Hand Rehabilitation," *2023 10th International Conference on Signal Processing and Integrated Networks (SPIN)*, pp. 538-542, Mar. 2023, doi: 10.1109/SPIN57001.2023.10117386
- [8] Amprimo G., Masi G., Pettiti G., Olmo G., Priano L., Ferraris C, "Hand tracking for clinical applications: Validation of the Google MediaPipe Hand (GMH) and the depth-enhanced GMH-D frameworks," *Biomedical Signal Processing and Control*, vol. 96, Oct. 2024, doi: 10.1016/j.bspc.2024.106508

- [9] M Sohan, T Sai Ram, R Reddy, C Venkata, "A review on yolov8 and its advancements," *International Conference on Data Intelligence and Cognitive Informatics*, pp. 529-545, Jan. 2024, doi: 10.1007/978-981-99-7962-2_39
- [10] Camillo Lugaresi, Jiuqiang Tang, Hadon Nash, Chris McClanahan, Esha Uboweja, Michael Hays, Fan Zhang, Chuo-Ling Chang, Ming Guang Yong, Juhyun Lee, Wan-Teh Chang, Wei Hua, Manfred Georg, and Matthias Grundmann, "MediaPipe: A Framework for Perceiving and Processing Reality," *Third workshop on computer vision for AR/VR at IEEE computer vision and pattern recognition (CVPR)*, vol. 2019, Jun. 2019
- [11] Wijaya, Ryan Satria, Santonius Santonius, Anugerah Wibisana, Eko Rudiawan Jamzuri, and Mochamad Ari Bagus Nugroho, "Comparative Study of YOLOv5, YOLOv7 and YOLOv8 for Robust Outdoor Detection," *Journal of Applied Electrical Engineering*, vol. 8, no. 1, pp. 37-43, Jun. 2024, doi:10.30871/jaee.v8i1.7207
- [12] Chitraningrum, Nidya, Lies Banowati, Dina Herdiana, Budi Mulyati, Indra Sakti, Ahmad Fudholi, Huzair Saputra, Salman Farishi, Kahlil Muchtar, and Agus Andria, "Comparison Study of Corn Leaf Disease Detection based on Deep Learning YOLO-v5 and YOLO-v8," *Journal of Engineering and Technological Sciences*, vol. 56, no. 1, pp. 29-40, Feb. 2024, doi: 10.5614/j.eng.technol.sci.2024.56.1.5
- [13] T. Sinkjaer, "Biopotentials as command and feedback signals in functional electrical stimulation systems," *Med Eng Phys*, vol. 25, pp. 29-40, Jan. 2003, doi: 10.1016/S1350-4533(02)00178-9
- [14] N. Bhagat, K. King, R. Ramdeo, A. Stein, and C. Bouton, "Determining grasp selection from arm trajectories via deep learning to enable functional hand movement in tetraplegia," *Bioelectron Med*, vol. 6, no. 1, pp. 17, Dec. 2020, doi:10.1186/42234-020-00053.5
- [15] Mohamad Hoda, Yehya Hoda, Abbass Hage, Abdulhameed Alelaiwi, and Abdulmoteleb El Saddik, "Cloud-based rehabilitation and recovery prediction system for stroke patients," *Cluster Computing*, vol. 18, pp. 803-185, Jun. 2015, doi:10.1007/s10586-015-0448-6
- [16] M. Jones, G. Collier, D.J. Reinkensmeyer, F. DeRuyter, J. Dzivak, D. Zondervan, and J. Morris, "Big Data Analytics and Sensor-Enhanced Activity Management to Improve Effectiveness and Efficiency of Out-patient Medical Rehabilitation," *International journal of environmental research and public health*, vol. 7, no. 3, pp. 748, Feb. 2020, doi:10.3390/ijerph17030748

