A Gesture-Based Multi-Agent Visual Learning Model for Acoustophoretic Interactions using a Swarm of AcoustoBots

PHCZ0
SEIoT MSc Final Project Report 2025
University College London
Supervisors: Sriram Subramanian, Narsimlu Kemsaram, Lei Gao

Abstract

This project presents the design and implementation of a real-time, camera-based gesture recognition visual learning model (VLM) for controlling a swarm of AcoustoBots—a multi-agent robotic system capable of haptic, audio, and levitation-based interactions through acoustic waves. The goal is to create an intuitive and contactless interface for users to switch between the three robot modalities using simple hand gestures. The novelty of this work lies in combining low-cost, camera-based gesture recognition with multimodal swarm robotics, enabling direct human-swarm interaction without the need for physical input devices.

The gesture recognition system is developed using CLIP-based VLM, a joint vision and language model that learns from mappings between images and text. CLIP is trained to associate visual features from images with corresponding textual descriptions, allowing it to classify images based on similarity to natural language labels. In this system, image frames are extracted from a live video feed and matched against predefined gesture prompts. The VLM enables zero-shot classification of distinct gestures, each mapped to corresponding AcoustoBot modalities.

The system evaluation is conducted in a controlled laboratory setting with one to two robots and users, using validation loss, accuracy, and latency as performance metrics. One of the most engaging aspects of this project was the immediate and visible robot response to human gestures—demonstrating a practical and compelling application of real-time human-robot interaction.

This project demonstrates the potential of using a lightweight, contactless vision-language approach to achieve intuitive human-swarm robot interaction. Through real-time gesture recognition, the system translates simple hand movements into immediate control commands that manipulate the behaviour of the multimodal AcoustoBots, allowing users to influence haptic, audio, and levitation responses. This enables seamless, real-time interaction between human intent and robot feedback. Future improvements will focus on scaling the number of AcoustoBots and supporting multi-user operations, enabling more complex coordination and broader applications in real-world environments.

Author Keywords
AcoustoBots, Gesture recognition, OpenCLIP, Swarm robotics, Human-robot interaction, Multi-agent systems, Visual Learning Model

Video Submission
Video of Experiment Demo

1 INTRODUCTION

The growing field of multi-agent systems emphasizes the development of swarm robotics that can coordinate to perform complex tasks in a distributed and adaptive manner. Swarm systems, inspired by biological collectives like ant colonies or bird flocks, are characterized by local sensing, decentralized control, and emergent global behavior [18]. Unlike traditional robotic systems that rely on explicit planning, swarm robotic systems are designed to operate using local rules, producing intricate global behaviors from simple agent interactions [16]. These characteristics enable the development of new robotics platforms with greater fault tolerance, scalability, and adaptability—traits that are vital for robotics applications in dynamic, real-world environments.

However, as swarm robotic platforms evolve beyond research prototypes into operational tools, a signifi-

MSC SEIoT FINAL PROJECT REPORT
Project report submitted in part fulfilment of the requirements for the degree of Master of Science System Engineering for the Internet of Things, University College London, 2025.

cant constraint remains in this field: How can humans intuitively interact with a collective of autonomous agents in real-time [24]? Traditional approaches to human-swarm interaction (HSI) often rely on abstract command languages, low-level input devices, or prescripted behaviors, which makes them inconvenient to non-expert users and impractical for fast-paced or dynamic environments [21]. An effective HSI system must strike a balance between control management and system autonomy, enabling the development of high-level abstraction layers that can translate human intent into swarm behavior.

Concurrently, recent advancements in visual-learning models (VLMs) offer compelling opportunities for building semantically aware perception systems. VLMs combine the representational power of deep convolutional neural networks with the interpretability of natural language models, allowing machines to not only "see" but also to "interpret" visual content in human-like ways. These models, such as CLIP (Contrastive Language-Image Pretraining) and Flamingo [36] [1], operate by embedding visual data and textual prompts into a shared latent space, enabling image classification. One of the most impactful capabilities of this architecture is zero-shot classification [39]. In traditional supervised learning, a model must be trained on a specific set of labeled examples for each class it is expected to recognize. This approach results in poor performance when the system encounters a new class, user, or environment not represented in the training data. VLMs, by contrast, can perform zero-shot inference. Given an image (e.g., a hand gesture) and a set of descriptive text prompts (e.g., "a closed fist," "an open palm," "a pointing finger"), the model can infer the correct label without having seen any labeled examples of that gesture during training. This generalization is possible because the model has been trained on a massive, diverse dataset of image–text pairs, often sourced from the web, allowing it to associate broad semantic concepts across both modalities. As a result, zero-shot VLMs can be flexibly adapted to new domains, vocabularies, and users without extensive training, making them particularly well-suited for dynamic and open-ended interaction settings, such as human-swarm interfaces.

Although most applications of VLMs have focused on single-robot structures, such as robotic manipulation, navigation, and embodied question answering, they have shown significant promise in multi-agent swarm systems. In contrast to traditional convolutional models, which depend heavily on labeled datasets hardcoded into the robot perception pipeline, VLMs can dynamically interpret semantic intent conveyed through language. This ability enables swarm robots to interpret user instructions more flexibly, reducing the need for retraining or domain-specific tuning. Recent works such as SwarmVLM [40] and ImpedanceGPT [4] highlight the growing interest in applying VLMs to swarm control and demonstrated that VLM can be implemented to modulate the collective behavior of distributed systems, allowing a human user to guide swarm robotics using descriptive commands rather than direct control input. Yet, these systems often assume language-based inputs from the user (e.g., text commands), which, while expressive, can be impractical in many deployment contexts. They lack a naturalistic input mechanism that connects human interaction with swarm systems in an intuitive and user-friendly way.

This is where gesture-based interaction presents an effective solution to the challenges of human-swarm interaction. Gestures represent a natural, non-verbal form of communication that humans frequently use in various contexts—from directing traffic to coordinating group tasks. They enable users to convey intent, commands, or feedback in a contactless and intuitive manner. As opposed to voice commands or tactile input, gestures require no explicit instrumentation and can be recognized using standard RGB cameras, making them ideal for deployment in low-cost distributed systems. Moreover, gestures do not require users to learn symbolic control languages or operate complex user interfaces, thus lowering the cognitive and training load on human operators and improving the general accessibility [27]. These features align well with the goals of human-swarm interaction, where the interface must balance responsiveness, intuitiveness, and scalability while still being deployable in real-world contexts.

This dissertation builds upon these themes and aims to design, implement, and evaluate a real-time gesture-based interface for swarm robots. The project leverages a vision-language model to classify user hand gestures captured by an embedded camera and translate them into multimodal actuation commands for a swarm of mobile acoustophoretic robots—AcoustoBots. These robots can deliver sensory feedback (audio, haptic, or levitation cues) via acoustic sound waves and are designed to operate cooperatively through centralized computation and local actuation. The proposed system uses low-cost ESP32-CAM to capture real-time video frames of hand gestures, which are then processed using OpenCLIP, a pre-trained vision-language model, to enable zero-shot gesture classification within the AcoustoBot control pipeline. Each gesture is mapped to a corresponding modality-switching command, allowing the user to switch the swarm's behavior between levitation, haptic feedback, and audio display. This gesture-driven interface is designed to be lightweight and adaptive, enabling intuitive human control over multi-agent robot systems without requiring manual retraining or complex equipment.

This work presents a novel approach to human-swarm interaction by developing a real-time gesture recognition system powered by vision-language models for semantic classification. By integrating low-cost hardware with a VLM inference pipeline, the system enables users to intuitively control the behavior of robot swarms through simple gesture inputs. The result is a scalable and adaptive interaction framework that supports embodied, human-centered control in dynamic real-world environments.

2 LITERATURE REVIEW

This section presents an overview of the relevant works across several research domains, including swarm robotics, visual learning models, gesture recognition, and acoustophoretic control systems. This review focuses on the theoretical foundations across these areas to identify existing capabilities, limitations, and key research gaps. The aim is to contextualise the novelty of this project and highlight the unique contribution to the development of AcoustoBots.

2.1 Multi-Agent and Swarm Robotics

Swarm robotics is a field that focuses on coordinating a large number of simple autonomous robots that communicate and interact locally with each other and their environment to exhibit emergent, collective behaviors. Inspired by natural systems such as ant colonies or bird flocks, swarm robotics emphasizes decentralized control, scalability, and robustness, allowing the group to perform complex tasks without centralized supervision.

Brambilia et al.[6] provided a foundational overview of swarm robotics engineering. Their research emphasises scalability, robustness, and flexibility as core principles for designing and implementing swarm systems. They propose a framework that categorises swarm control strategies into three approaches: behaviour-based control, bio-inspired methods, and formal model-driven approaches. The paper underlines common challenges in the task, including behaviour emergence, fault tolerance, and real-time adaptability. One of their key findings is that local rules, when properly designed, can scale effectively to large numbers of agents without centralised coordination. Their work demonstrates that complex global behaviour can emerge from simple decentralised local interactions—a principle used in this project's swarm control architecture.

Building on these principles, recent advancements by Ichihashi et al. present Swarm Body[18], a system that explores the idea of embodied swarm interaction. Unlike traditional multi-agent systems that focus on abstract collective goals, Swarm Body executes in direct, tangible movement with swarm formations. Their experiments demonstrate that users can manipulate swarm configurations in real-time using spatial and gestural cues, effectively treating the swarm as a physical interface and transforming it from an abstract distributed system into direct physical interaction with human users. However, these systems typically lack seamless input for dynamic human control. This dissertation builds on the findings of Ichihashi et al. by proposing a more autonomous form of interaction using camera-based hand gesture recognition powered by a vision-language model. This allows users to control swarm modalities in real-time through semantically grounded, contactless gestures.

Although decentralization is widely regarded as a defining principle of swarm robotics, this project adopts a centralized control strategy. This choice is motivated by the computational constraints of the AcoustoBots' onboard microcontrollers, which lack the capacity to independently execute the resource-intensive visual language model required for gesture recognition. Offloading computation to a central processing server ensures consistent real-time performance in gesture classification and enables reliable coordination between robots and users. While this approach departs from the ideal of fully distributed swarm autonomy, it constitutes a practical design compromise that aligns with the project's goal of achieving real-time camera-based gesture interaction. Future work may explore decentralized control using advanced embedded hardware.

2.2 Acoustic Levitation and Multimodal Swarm Interaction

Acoustophoresis is a technique that uses acoustic radiation pressure from sound waves to suspend and manipulate particles, cells, or even liquids. This phenomenon was first observed by Kundt in 1866 through his experiment with standing waves in a tube and later studied in greater depth by King in 1934 [22]. King explains that acoustic levitation occurs when high-frequency ultrasonic waves interfere to form standing wave patterns. At regions of minimal acoustic pressure known as pressure nodes, objects would experience an upward force that counteracts gravity, enabling levitation. By adjusting the phase and amplitude of these waves, acoustic traps can be generated and moved, allowing for precise, contactless manipulation of small particles in air, liquid, or biological environments.

Acoustophoretic systems have traditionally been static, relying on large phased arrays mounted in fixed positions. Early studies emphasise piezoelectric transducers paired with reflectors to form standing waves capable of handling and transporting small particles in mid-air [13]. Later, systems like TinyLev introduced multiple emitters arranged in hemispherical configurations, enabling stable acoustic traps without the need for precise mechanical alignment [30]. Recent advancements in phased array transducers have further improved manipulation precision and spatial control. Melde et al. demonstrated the use of ultrasonic holograms to shape complex 3D acoustic fields for particle manipulation [31]. In their detailed overview, Andrade et al. [2] analysed several acoustic levitation platforms for micro and mesoscale applications and identified key technical constraints relevant to mobile acoustic robot systems. They noted that precise phase and amplitude control are essential when manipulating various materials, shapes, and sizes. They also highlighted that environmental changes such as temperature variations and acoustic reflections could reduce trapping stability, suggesting the need for intelligent recalibration or sensor feedback in mobile deployments. Adjustments to the array geometry, transducer spacing, and frequency can all lead to improvement of trapping performance.

A milestone in the work of the acoustic levitation is the introduction of GPU-accelerated phase retrieval algorithms, notably GS-PAT (Gerchberg–Saxton Phased-Array Transducers) developed by Martínez Plasencia et al. [33]. The innovation facilitated the computation of multi-point ultrasonic fields at extremely high rates, reaching up to 17kHz for 32 simultaneous focal points. This capability supports new applications in haptics, levitation, and audio through rapid multi-point steering. GS-PAT works by iteratively solving the phase retrieval problem using a modified Gerchberg–Saxton algorithm [14] optimised for phased arrays. It utilises GPU-accelerated matrix operations to compute phase and amplitude settings across all transducers in parallel. As a result, it can rapidly generate complex acoustic holograms for multiple points in real time—an essential enhancement compared to previous techniques limited to single-point or slow updates. GS-PAT is directly employed in the AcoustoBots system, powering their phased array modules. Each robot uses GS-PAT-generated beamforming parameters to dynamically steer acoustic waves for haptics, audio, or levitation, without the need for mechanical repositioning. This enables real-time multi-point control, such as simultaneous tactile feedback at multiple locations or coordinated particle levitation. The implementation of GS-PAT enhances AcoustoBots' capabilities by enabling rapid reconfiguration of focal points, which is vital for multimodal swarm interaction scenarios where timing and spatial precision are critical.

The AcoustoBots system introduced by Kemsaram et al. represents a significant advancement in acoustophoretic robotics by integrating adjustable mobile ultrasonic phased arrays into individual robots, enabling them to deliver mid-air haptic feedback, directional audio, and acoustic levitation [19]. These three capabilities are illustrated in Figure 1. Each AcoustoBot is built on a Mona robot base, a small wheeled robot with onboard power and wireless communication. Mounted on top is a custom-designed 8×8 ultrasonic phased array transducer (PAT) board, attached to a motorised hinge that allows precise angular control of the sound field. This layout enables each robot to redirect its acoustic output in three-dimensional space without physically repositioning itself.

The core of this system relies on the acoustic radiation force, produced by the interference of ultrasonic waves. By manipulating phase and amplitude across the transducer array, the robots can shape pressure fields that can be felt (haptic feedback), heard (through parametric audio), or used to levitate lightweight particles (acoustic levitation). Control signals are handled by a central processing server, which uses a physical simulation to determine beamforming parameters. These parameters are transmitted via UDP to each robot's ESP32 microcontroller, which relays the signal to a field-programmable gate array (FPGA) responsible for driving the transducer array. The hinge actuator adjusts the PAT orientation based on commands from the same server, enabling spatial targeting of the acoustic output [19].

AcoustoBots are designed to be modular and scalable. Since each robot is mobile and can function individually, the system supports dynamic reconfiguration of swarm formations. Additional robots can increase coverage, distribute modalities over larger spaces, or perform more complex coordinated behaviours, such as synchronised levitation or multi-user interaction zones.

However, a significant limitation of the current AcoustoBot system lies in its lack of an intuitive user control interface. While the central server orchestrates low-level actuation and hinge control, there is no natural real-time mechanism for user input. Existing demonstrations rely on scripted commands or preprogrammed configurations, limiting adaptability and human interaction. This project addresses that gap by introducing a vision-based gesture recognition interface as a control layer on top of the AcoustoBot system. By using a CLIP-powered visual learning model, the system allows users to activate modality-switching commands (i.e. haptic, audio, or levitation) through predefined hand gestures. This approach transforms

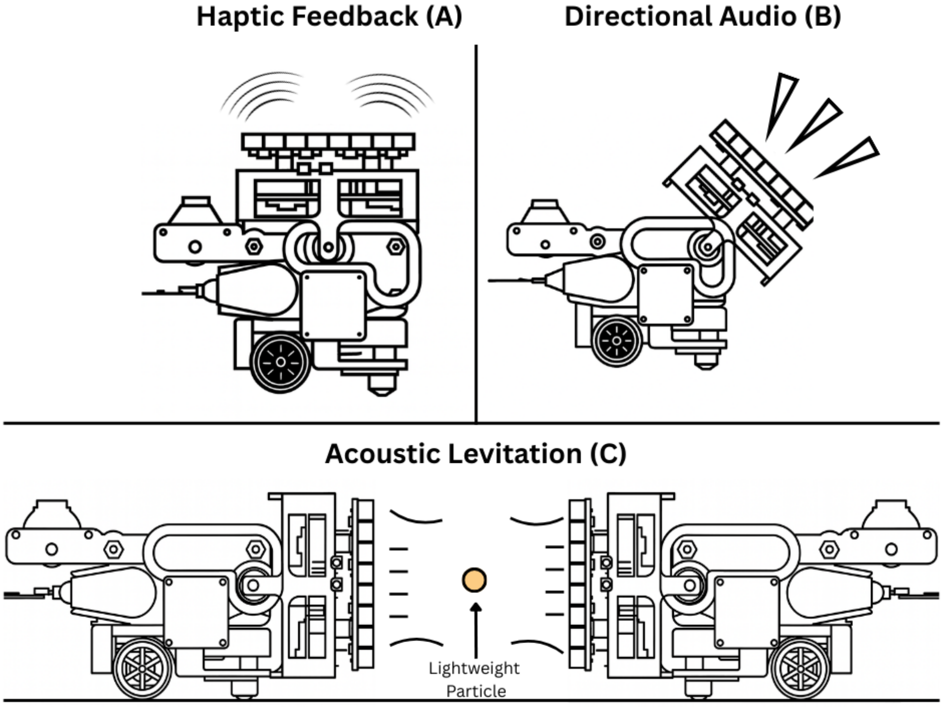

Figure 1: The AcoustoBot system demonstrates three key functions enabled by its Phased Array Transducer: (A) mid-air haptic feedback, (B) directional audio projection, and (C) acoustic levitation of lightweight particles.

AcoustoBots from a programmable hardware platform into a responsive multimodal system that supports contactless real-time human-swarm interaction.

2.3 Visual Learning Models

Conventional image classifiers usually require fully labelled datasets and are constrained to a predefined, fixed number of classes. These models, often built using convolutional neural networks (CNNs), perform well when trained on large, curated datasets such as ImageNet [11]. However, they struggle to generalise to new categories without retraining and lack flexibility in open-ended classification tasks. Zhang et al. further demonstrated that deep networks commonly memorize training sets rather than learn robust features, resulting in poor generalization [41]. Recent work on CNN behavior with imbalanced datasets has also shown that rare classes (or uncommon gestures) receive lower generalization performance [10]. These systems often require extensive preprocessing and are not easily adaptable to new gesture sets or deployment environments without additional supervised learning steps[34]. Overall, these findings highlight that traditional CNN-based classifiers, while powerful in controlled settings, struggle in open-ended real-world gesture recognition scenarios, underlying the need for more flexible and robust alternatives.

To address these challenges, recent studies have utilised deep learning and transformer-based techniques for hand gesture recognition. Visual-Language Models (VLMs) have emerged as a powerful model that learns from image and text data simultaneously, enabling cross-modal understanding. These models are typically built on paired datasets and trained to align visual and textual representations in a shared embedding space. Over the last few years, numerous VLMs have been developed, each optimised for different tasks and deployment contexts.

One such model is ALBEF (Align Before Fuse), proposed by Li et al. [26]. ALBEF has a two-stage training process that first aligns unimodal (visual and textual) embeddings through contrastive learning before fusing them for downstream tasks like image-text retrieval and visual question answering (VQA). ALBEF performs strongly in vision-language reasoning tasks but requires task-specific fine-tuning, making it less ideal for general-purpose or real-time inference applications

Another important model is Flamingo, developed by DeepMind [1], which combines frozen vision backbones with large language models to perform few-shot learning across multiple tasks. Flamingo demonstrates strong generalisation capabilities and excels in open-ended tasks such as image captioning, video understanding, and interactive multimodal reasoning. However, it is designed primarily for large-scale research settings because it requires significant computational power, limiting its practical use in embedded systems.

While these models offer powerful multimodal reasoning and generative capabilities, their design emphasises flexibility over efficiency. CLIP, introduced by Radford et al. [36], is a general-purpose VLM trained on 400 million (image, text) pairs using a contrastive learning objective that aligns visual and textual embeddings in a shared latent space. CLIP was designed for zero-shot classification: given a text prompt and an image, the model determines which descriptions best match the image without specific training on those classes. This feature removes the need for gesture-specific training and supports real-time execution. OpenCLIP [9] extended this work by reproducing and scaling CLIP using open datasets. They presented scaling laws that show improvements in performance as model size and dataset scale increase. OpenCLIP's flexibility, reproducibility, and open-source models make it ideal for real-world integration. This project implements OpenCLIP for gesture recognition, using its multimodal embedding space to classify hand gestures by associating them with textual descriptions such as "fist", "thumbs up", and "palm".

2.4 Gesture Recognition and Human-Robot Interaction

Gesture recognition in the context of human-robot interaction (HRI) enables users to communicate with robots through natural, contactless hand movements. This form of interaction is exceptionally valuable in dynamic or constrained environments where traditional input methods like keyboards, joysticks, or touchpads are impractical. In HRI systems, gestures can be applied for tasks ranging from navigation commands and object manipulation to system control. Gesture-based control is particularly effective for multimodal systems, such as the AcoustoBots, where different gestures can intuitively switch between functional modes.

Several studies have explored vision-based gesture recognition for robot control. Oudah et al. [32] present a foundational review of camera-based gesture recognition techniques, such as skin-colour segmentation, contour-based modelling, and depth imaging, while highlighting trade-offs in lighting, computational efficiency, gesture complexity, and accuracy across various applications. This research shows that while traditional methods can be effective, they often struggle with variations in lighting, background clutter, or generalising across users and environments. Furthermore, Qi et al. [34] provide a comprehensive review of computer vision-based hand gesture recognition in HRI systems, categorising approaches into static and dynamic gesture models. Static gestures involve fixed hand shapes and are typically classified using image-based features such as hand contour, skin colour, or hand keypoints. Dynamic gestures, involving motion trajectories (e.g., waving), often require temporal models like recurrent neural networks (RNNs) or 3D convolutional networks.

Malobický et al. [29] present a practical HRI system that combines vision-based gesture recognition and tool handover. Their work demonstrates how computer vision can facilitate seamless collaboration between humans and robots in a shared workspace. Their method uses a YOLO-based object and gesture detection and primarily focuses on manipulation tasks rather than swarm or modality control. In contrast, our project uses gesture recognition to switch between swarm-level behaviours in a multimodal robotic system rather than interaction between individual objects.

Tan et al. [37] underscores a crucial challenge in gesture-based HRI: the lack of a standardised gesture vocabulary. Through user testing, they demonstrate that uninstructed gestures vary widely among users, supporting the need for a predefined gesture set. They propose a set of communicative gestures and develop a gesture recogniser based on RGB input integrated into ROS. This directly relates to our project, which focuses on defining three distinct static gestures to ensure consistency and reduce ambiguity during real-time interaction.

In summary, while previous research has established a solid foundation for gesture-based interaction in HRI, most approaches are constrained by dataset-specific models, limited generalisation, or lack of integration with multimodal swarm systems. This project introduces a semantic, real-time gesture interface that enhances the capabilities of AcoustoBots by providing an intuitive and adaptive control mechanism for users to engage with swarm robots in an interactive environment.

3 SYSTEM OVERVIEW AND ARCHITECTURE

This section describes the architecture and operational flow of the gesture-controlled AcoustoBot system developed in this project. The system diagram is illustrated in Figure 2. The system enables real-time contactless interaction between users and AcoustoBots through the use of gesture recognition and motion tracking technologies. Each robot is paired with a user and responds to both their position and semantic gesture commands. The system integrates several hardware and software components, including ESP32-CAM modules, a PhaseSpace motion capture system, a central server, an OpenCLIP-based gesture VLM,

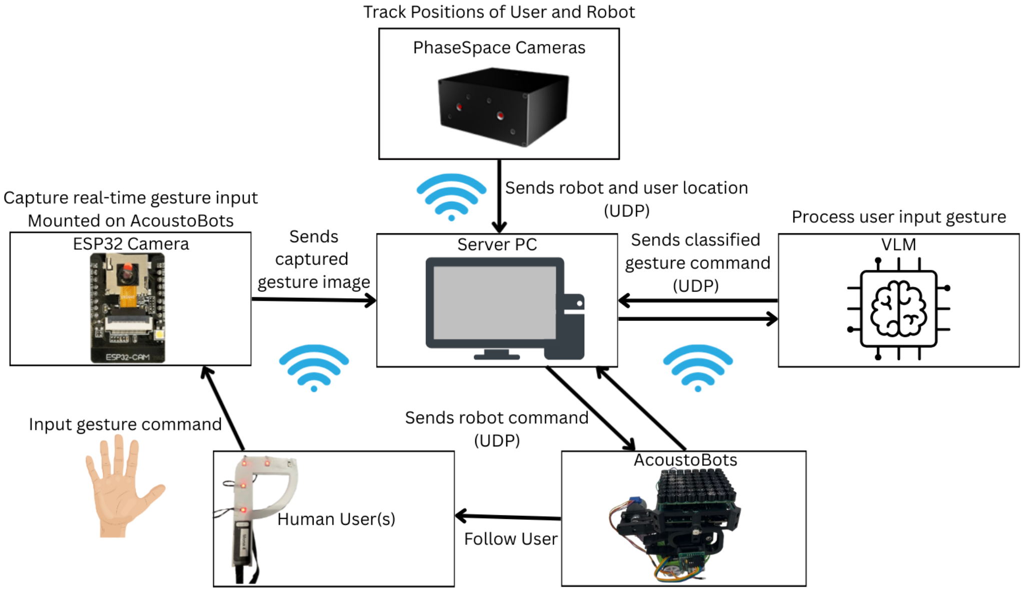
Track Positions of User and Robot
PhaseSpace Cameras
Sends robot and user location (UDP)
Capture real-time gesture input
Mounted on AcoustoBots
ESP32 Camera
ESP32-CAM
Sends captured gesture image
Server PC
Sends classified gesture command (UDP)
Process user input gesture
VLM
Input gesture command
Sends robot command (UDP)
AcoustoBots
Human User(s)
Follow User

Figure 2: Gesture Interaction System Diagram and Data Flow

and the AcoustoBot platform.

3.1 PhaseSpace Motion Tracking System

To enable robots to follow users autonomously, the system employs the PhaseSpace Impulse X2 motion capture system for precise real-time position tracking. The PhaseSpace system uses LED makers that emit structured infrared pulses, which are captured by the PhaseSpace cameras and processed to determine the six degrees of freedom position and orientation of each object. Each AcoustoBot is equipped with LED markers mounted on its chassis, while users carry a joystick-like handle embedded with similar LED markers, as shown in Figure 3. This setup allows the system to associate each user with a specific robot, ensuring that tracking and control are one-to-one.

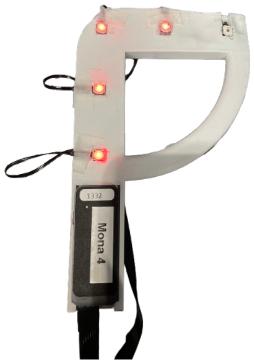

Figure 3: User PhaseSpace LED Handle

The PhaseSpace system continuously tracks the three-dimensional position and orientation between each user and their assigned robot at high update rates, producing real-time motion data. This position data is then transmitted to the central server via UDP, ensuring minimal communication delay and enabling immediate responsiveness.

The server then receives the positional data of each object and processes it to issue movement commands for the robot. Control algorithms generate motion commands to adjust the robot's heading, speed, and distance so that it can follow its designated user in real-time while maintaining a collision-free movement. This continuous position-feedback loop forms the foundation for the spatial interaction between the swarm robot and the user in our system.

3.2 ESP32-CAM for Gesture Input Capture

Each AcoustoBot is equipped with an ESP32-CAM, a microcontroller unit with integrated Wi-Fi functionality and a low-resolution camera module. The purpose of the camera is to capture real-time video frames of hand gestures, enabling contactless user input. This design addresses the research goal of creating a low-cost, contactless interface for modality control without additional sensors or physical controllers. By distributing cameras across the swarm, each robot can independently sense and respond to user commands, providing a foundation for scalable multi-robot interaction.

The camera continuously transmits frames over Wi-Fi to the central server PC for processing. This setup ensures that the robots remain lightweight, with only the necessary sensing hardware on board, while the computationally intensive recognition tasks are handled centrally. Captured frames are transmitted directly via UDP to the central server. Upon receiving the video stream, the server preprocesses each frame by resizing it to prepare the image for input into the Visual-Language Model. This centralised structure ensures that the system achieves real-time responsiveness and maintains a high level of recognition accuracy, two crucial requirements for interactive human-swarm robotics.

However, this choice embodies an essential design trade-off. While central processing reduces computational demand from the robots, it contradicts the long-term vision of decentralised swarm robotics. Ideally, each AcoustoBot should be capable of onboard inference, running gesture recognition directly on its microcontroller to eliminate reliance on a central server. However, this is currently infeasible, as the visual learning model requires significant processing power and memory that far exceeds the capabilities of a simple ESP32 microprocessor. As a result, the system prioritises functionality and feasibility over decentralisation.

3.3 Visual Learning Model for Gesture Classification

The core of our system is the Visual Learning Model, which plays a central role in bridging user intention and robotic actuation within the system architecture. The primary responsibility of the VLM is to interpret the visual input captured by the ESP32-CAm and transform it into functionality commands for the AcoustoBots. The VLM allows the system to operate as an intuitive, contactless interface where user gestures directly control robot behaviour.

The workflow of the VLM begins when the ESP32-CAM captures the user's real-time gesture input. The captured video frames are transmitted to the Server PC, which forwards the data to the VLM for interpretation. The VLM then classifies the gesture into one of the predefined gesture sets. Instead of remaining at the level of raw image data, the VLM converts the gesture into a discrete command signal, such as “haptic mode”, “levitation mode”, or “audio mode”.

Once the gesture is classified, the VLM sends the recognised command back to the Server PC, which is integrated with additional contextual data, such as user and robot positions tracked by the PhaseSpace system. The server then combines these inputs to generate control commands to the AcoustoBots. This design ensures that the robot not only reacts to gestures but also executes the command in the correct spatial context, for example, following its designated user while simultaneously adjusting its modality.

In our system architecture, the VLM acts as the interpretation layer between human users and the swarm robots. By converting visual input into well-defined semantic commands, it reduces complexity and ensures that the user's objective is accurately conveyed to the robots. Its integration into the workflow exemplifies the design principle of making swarm robotics more conventional, scalable, and interactive through a natural and efficient interface.

3.4 AcoustoBot Response and Multimodal Actuation

The physical layer of the system workflow lies in the AcoustoBots, which execute the control commands produced by the gesture classification and server integration. Each AcoustoBot is equipped with an ultrasonic phased array transducer (PAT) board, mounted on a motorised hinge, which allows the robot to project acoustic wave fields in three-dimensional space. Depending on the gesture recognised by the VLM, the robot switches between three distinct modalities: haptic, levitation, and audio. For instance, an open palm gesture activates haptic feedback, a closed fist triggers directional audio output, and a thumbs-up initiates levitation.

This multimodal actuation allows physical interaction between the user and the system by directly linking the high-level semantic classification performed by the VLM to low-level robotic control. From a research perspective, the AcoustoBot response layer offers new insights into the swarm system. First, it demonstrates that gesture-based commands can extend beyond navigation and into the control of multimodal robotic functions, broadening the scope of human-swarm interaction. Second, it highlights the trade-off between centralised and decentralised control. While the current system relies on a central server to issue actuation commands, the design shows the feasibility of integrating multiple modalities in a scalable swarm setting. However, it also underlines a limitation: the lack of distributed autonomy in individual robots, which prevents the system from fully achieving decentralisation. In terms of evaluation, the AcoustoBot actuation layer provides clear performance metrics, such as the accuracy of modality switching, response latency, and user feedback.

3.5 Server and Communication Architecture

The central server coordinates communication among the various system components, processes and integrates inputs from multiple sources, and subsequently issues control commands to the AcoustoBots. In our system architecture, all data streams, including user position tracking from the PhaseSpace camera, video frames of gesture input from the ESP32-CAM, and classification results from the visual language model, are processed on the server PC. The centralised design improves consistency by maintaining a unified representation of both user intent and robot state across the entire system. It also enhances efficiency, since computationally intensive tasks such as gesture classification are executed centrally rather than being redundantly processed on individual robots. Furthermore, centralisation strengthens system-level control, enabling coordinated responses that allow the swarm to operate smoothly and predictably under dynamic conditions.

Communication between the server and distributed hardware units is established over a Wi-Fi network using the User Datagram Protocol (UDP). UDP is selected for its low latency, making it well-suited for real-time robotic applications where responsiveness is prioritised over guaranteed packet delivery. Each ESP32-CAM module transmits captured gesture frames to the server via UDP packets, while the server returns processed classification results and control signals to the AcoustoBots using the same protocol. In parallel, position data from the PhaseSpace tracking system is streamed in real time, enabling the server to compute and forward robot-following commands with minimal delay.

A key limitation of this centralised communication architecture is its deviation from the principle of decentralisation in swarm robotics, where individual agents operate autonomously with minimal reliance on a global controller. While centralisation enhances efficiency and feasibility, it does so at the cost of scalability and fault tolerance—representing an explicit design trade-off that also highlights potential directions for future research.

4 GESTURE RECOGNITION MODEL

4.1 OpenCLIP Architecture

Our Visual Learning Model is built upon OpenCLIP, an open-source implementation of the Contrastive Language-Image Pre-training (CLIP) architecture developed by OpenAI [36]. OpenCLIP provides a foundation model that has been pre-trained on large-scale datasets containing image-text pairs, enabling it to learn visual representations that are semantically meaningful across different domains.

For this project, the model employs a Vision Transformer (ViT) backbone, an architecture that has gained widespread adoption in computer vision due to its ability to capture long-range dependencies in images more effectively than traditional convolutional neural networks (CNNs). Instead of processing images pixel by pixel or with fixed convolutional kernels, the ViT divides each input image into small, non-overlapping patches. Each patch is then "flattened" into a vector and embedded in a sequence, which is processed through multiple layers of a transformer network [12]. This enables the model to build a hierarchical representation of the image, capturing both local details, such as the shape of a hand or orientation of fingers, as well as global context like the overall gesture pose.

The pre-training process of OpenCLIP involves contrastive learning, where the model learns to associate visual representations with corresponding textual descriptions. During this process, both images and text are encoded into a shared embedding space, where semantically related image-text pairs are positioned closer together, and unrelated ones are further apart. The training objective is to maximise the similarity between matching image-text pairs while minimising similarity between non-matching pairs [8]. Contrastive learning enables the model to develop a unified representation space where semantically similar notions are positioned closer together, regardless of whether they are visual or textual information. The resulting visual encoder produces 512-dimensional feature vectors that capture high-level semantic information about the input images.

By leveraging this architecture, OpenCLIP offers a robust framework for mapping gesture images captured by the ESP32-CAM into the same semantic space as natural language gesture descriptions. The server can then classify gestures by comparing the image embeddings against the embeddings of these textual labels, selecting the closest match. This approach eliminates the need for gesture-specific retraining, therefore enhancing system flexibility and adaptability, while maintaining the low-latency performance required for real-time control of the AcoustoBots.

4.2 Feature Extraction and Linear Probing

The novelty of our VLM system lies in its approach to leverage pre-trained visual representations for gesture recognition without retraining the entire model. The approach follows a two-stage procedure. First, the pre-trained CLIP visual encoder is frozen and employed solely as a feature extractor. Input images are preprocessed—resized to 224x224 pixels, normalised according to ImageNet statistics, and converted to tensor format [17] —before being passed to the encoder. This produces a 512-dimensional feature vector that captures the high-level semantic features of the gesture, enabling accurate classification with minimal additional training.

These extracted features serve as a semantically meaningful representation that captures the characteristics of hand gestures. The advantage of using a pre-trained CLIP model is that it encodes general visual knowledge about shapes, textures, and spatial

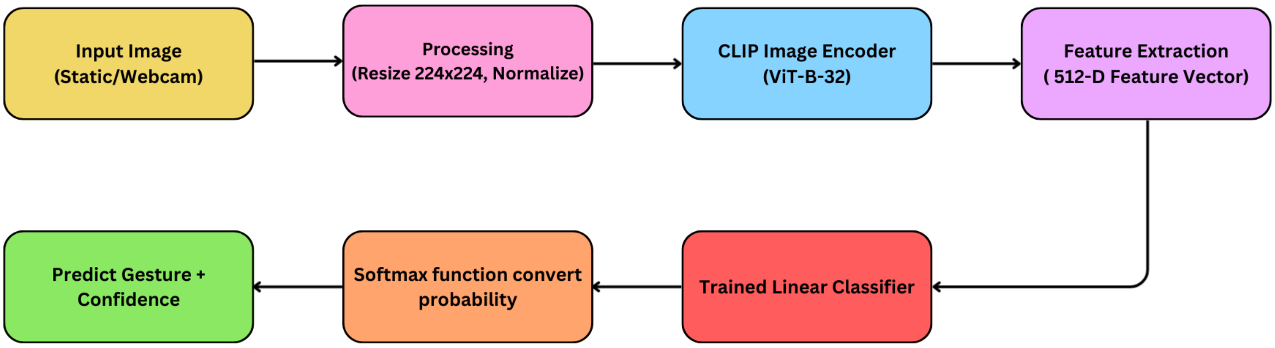

Figure 4: Inference phase workflow for gesture recognition. Input images are preprocessed, encoded using the frozen CLIP image encoder, and classified by the trained linear classifier. Softmax activation produces class probabilities, from which the final predicted gesture is selected.

relationships that are transferable to the specific domain of gesture recognition [42]. The 512-dimensional feature space has the ability to represent the complex visual patterns associated with different hand configurations while maintaining computational efficiency.

The second stage of the procedure employs a linear probing approach, where a simple linear classifier is trained on top of the frozen CLIP features to perform specific gesture classification tasks. This approach relies on the principle that pre-trained representations often contain sufficient information for downstream tasks, requiring only a minimal adaptation layer rather than full fine-tuning [25]. In our case, the linear probe is implemented as a single fully connected layer that maps the 512-dimensional CLIP embeddings into three discrete gesture categories: thumbs up, fist, and palm.

The linear probing approach offers several advantages over full fine-tuning. First, it preserves the general visual knowledge encoded in the CLIP model, thereby preventing catastrophic forgetting—a phenomenon in artificial neural networks where a model abruptly forgets previously learned information after learning a new task [23]. Second, it significantly reduces the number of trainable parameters, making the training process more computationally efficient and less prone to overfitting, especially with limited training data [35]. Finally, it enables rapid adaptation to new gesture categories, as only the lightweight classifier must be retrained while the core encoder remains fixed, enhancing the flexibility of the model.

4.3 Dataset Construction

The training dataset for our project is constructed through a structured image collection and annotation process. The dataset comprises three categories: thumbs up, fist, and palm. Each category contains hundreds of images captured under various conditions. While some of the dataset was captured directly using our lab camera, most images were sourced from publicly available online databases for gesture recognition. This approach ensured that the dataset incorporated a broad variety drawn from diverse environments and users, strengthening the model's generalization ability. The dataset includes variations in lighting conditions, background complexity, and hand orientation to further enhance representativeness. This diversity is crucial for training a robust model that can perform reliably in real-world scenarios under different environments and contexts [38].

The annotation process involves labelling each image with its corresponding gesture category, creating a tab-separated annotation file that maps image filenames to their labels. This consistent labelling schema reduces ambiguity in category assignment and facilitates reproducibility in dataset preparation during model training. By establishing a well-defined taxonomy of gestures, the dataset provides a solid foundation for supervised learning.

Moreover, the dataset construction addresses the lack of domain-specific resources for gesture recognition. While large-scale vision-language models such as CLIP are trained on general-purpose image-text corpora, they often require adaptation to specialised inputs. By curating a diverse, annotated dataset focused exclusively on hand gestures, this project enables efficient adaptation of pre-trained models, reinforcing the system's robustness in dynamic human-robot interaction scenarios.

4.4 Training Methodology and Loss Function

The training process of the visual learning model follows a supervised learning paradigm where the system learns to associate visual features with discrete gesture labels. This methodology is illustrated in Figure 5. To ensure robust evaluation, the dataset is divided into an 80/20 train-validation split, where 80% of the

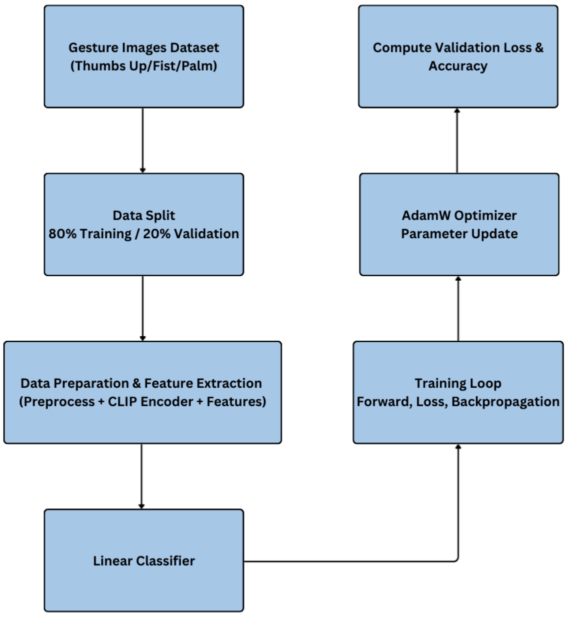
Gesture Images Dataset (Thumbs Up/Fist/Palm)
Data Split 80% Training / 20% Validation
Data Preparation & Feature Extraction (Preprocess + CLIP Encoder + Features)
Linear Classifier
Training Loop Forward, Loss, Backpropagation
AdamW Optimizer Parameter Update
Compute Validation Loss & Accuracy

Figure 5: Training Phase Workflow for the Gesture Classification Model

data is used for train the model and the remaining 20% is reserved for validation. This setup ensures that the model performance is regularly assessed on unseen data, reducing the risk of overfitting during training.

The optimisation process employs the AdamW algorithm with a learning rate of 1e-3. AdamW optimiser combines the adaptive learning rate adjustment of the original Adam algorithm with decoupled weight decay, which helps to control overfitting and improve convergence stability [28]. AdamW is particularly effective in deep learning applications with sparse or unstable gradients. The model uses Cross-Entropy Loss as the objective function, a standard approach for multi-class classification problems that provides stable gradients and reliable feedback during training [15].

The training process consists of 50 epochs with a batch size of 5, chosen to strike a balance between computational efficiency and gradient stability. While larger batch sizes may speed up training, they often lead to poorer generalisation. In this instance, a smaller batch size is appropriate for the dataset size and helps prevent overfitting while maintaining stable training dynamics [20]. During each epoch, the training data is passed through the network, gradients are computed, and only the parameters of the linear probe are updated, while the CLIP encoder remains frozen. This design ensures that the model benefits from the pre-trained features of CLIP while efficiently adapting to the specialised task of gesture recognition.

4.5 Model Exportation and Deployment

Upon completion of training, the system saves the trained linear probe model using PyTorch's state dictionary format, which preserves the optimised weights and biases in a platform-independent manner. The system also updates the class mapping file, which maintains the bidirectional relationship between numerical indices and gesture class names, ensuring consistency between the training and inference phases.

The trained model is deployed through an inference pipeline for real-time gesture recognition. The inference process begins with image preprocessing, where the input frames are resized, normalised, and converted to the appropriate tensor format. This step ensures consistency with the training pipeline and prepares the data for feature extraction.

The preprocessed image is then fed through the frozen CLIP encoder, which generates a 512-dimensional feature vector consisting of the semantic characteristics of the gesture frame. These extracted features are subsequently fed into the trained linear probe model, which outputs a set of raw values, known as logits. Logits can be understood as the unnormalised confidence score of the classification model; they may be positive or negative and do not have any probabilistic meaning yet. Each gesture class has its own corresponding logit. The system applies the softmax function to convert these logits into interpretable probabilities, which normalises the values so that they lie between 0 and 1 and sum to 1 [7]. The system selects the gesture class with the highest probability as the predicted output, and the corresponding probability value is used as a confidence score that reflects the model's certainty in its decision.

To support deployment beyond the training environment, the system incorporates functionality for exporting the trained classification model to the ONNX (Open Neural Network Exchange) format [3]. The ONNX export process integrates the CLIP encoder and the linear probe into a single computational graph, allowing the entire inference pipeline to be executed as one unified model. This integration avoids the need for separate model loading or feature extraction steps during deployment. By adopting ONNX, the system achieves platform independence, optimized inference performance, and compatibility with various deployment frameworks. Notably, ONNX exportation enables the model to run seamlessly on Windows platforms, where the AcoustoBot application is executed. The ONNX exportation ensures that the gesture recognition model can be directly integrated into the centralised control server, maintaining consistency and efficiency across the entire system.

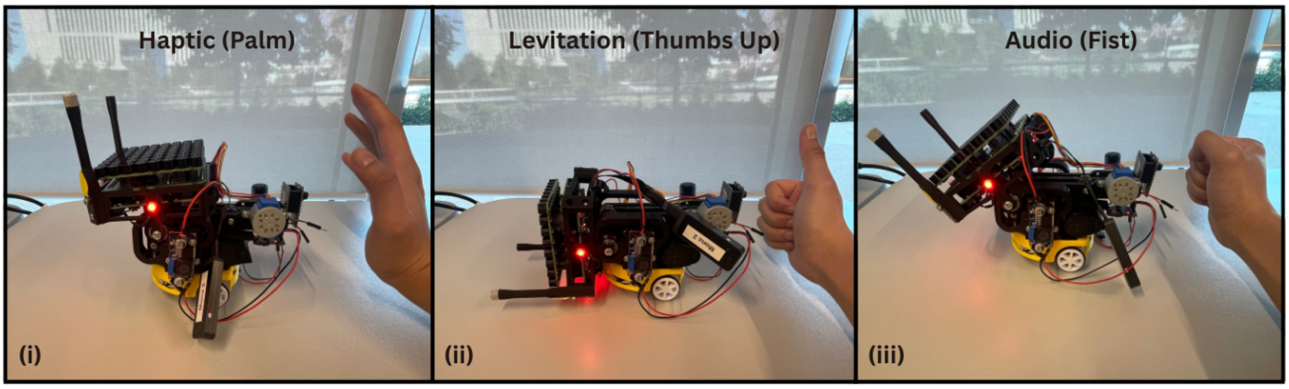

Figure 6: Gesture-to-Modality Mapping for AcoustoBot Control: (i) Open Palm triggers **Haptic Mode**, (ii) Thumbs Up triggers **Levitation Mode**, and (iii) Fist triggers **Audio Mode**. Each gesture is captured by an ESP32-CAM and interpreted using a vision-language model to activate the corresponding multimodal actuation.

4.6 Integration with AcoustoBot Control

The gesture recognition system is designed to integrate seamlessly with the Acoustobot control architectures, where each recognized gesture maps to specific robot modalities. The system outputs discrete gesture classifications and passes the result to the Acoustobot control software, enabling intuitive human-robot interaction through natural hand gestures. The real-time nature of the inference pipeline ensures that gesture commands can be processed with minimal latency, supporting responsive robot control.

The three-gesture classification model provides a sufficient vocabulary for basic robot control while maintaining simplicity and reliability. In our system, the thumbs up gesture is mapped to acoustic levitation, the fist gesture enables audio playback, and the palm gesture activates haptic feedback. Figure 6 shows the three hand gestures employed to control AcoustoBot behaviors, with each gesture corresponding to a distinct actuation mode. This mapping strategy creates an intuitive interface where users can communicate their intentions to the robot through natural hand movements, reducing the need for complex programming or remote control interfaces.

To improve robustness, the system incorporates confidence scores generated during classification. The confidence score is treated as a rejection threshold, allowing the system to discard low-confidence predictions and thereby improving the robustness of the gesture recognition process in dynamic human–robot interaction contexts. So if the confidence score of the predicted gesture falls below the defined threshold, the central server does not transmit control commands to the robot. This threshold mechanism prevents unintended or erroneous actuation caused by uncertain predictions, ensuring that the robot only responds when the system is sufficiently confident, aligning robot behaviour with the user's intentions.

The technical architecture demonstrates how efficient transfer learning techniques can effectively adapt pre-trained vision-language models for specific robot applications. By combining gesture classification with a confidence-based threshold, the system delivers robustness and real-time responsiveness, ensuring accurate control while maintaining computational efficiency.

5 EVALUATION

The evaluation of our proposed system is conducted in two parts to assess both the performance of the gesture classification model and its integration within the AcoustoBot platform. The first part focuses on evaluating the visual learning model, examining metrics such as training loss, validation loss, and classification accuracy to determine the model's effectiveness in recognising input gesture images. The second part evaluates the end-to-end integration of the model with the AcoustoBot system through a series of experimental setups. This includes testing the real-time gesture recognition pipeline in conjunction with robot actuation, measuring the response time and accuracy of modality switching. Together, these evaluations provide a comprehensive assessment of the system, ensuring both the technical validity of the model and its practical effectiveness in enabling intuitive real-time human-swarm interaction in real-world scenarios.

5.1 Evaluation of the Gesture Classification Model

The performance of the gesture classification model is first assessed independently. The dataset was divided into an 80:20 training-to-validation split to analyse the generalisation capability of the trained model. The evaluation process incorporates the validation set, which is held out from the training data and used to monitor the model's performance on unseen samples after each training run. Training and validation loss are recorded over 50 epochs to evaluate convergence behaviour and potential overfitting.

The evaluation of the gesture classification model was conducted by recording a total of 8 training runs, each using a different dataset size. This approach systematically compared how dataset scale influences generalisation performance and provided valuable insights into the effectiveness of the visual learning model. To assess model performance, training loss, validation loss, and validation accuracy were used as the primary metrics, with training speed also compared across the different runs.

5.1.1 Training Speed

As shown in Figure 7, the training speed, measured as epoch duration, remained consistently low across all eight runs regardless of dataset size. The initial epoch required slightly longer processing time due to model setup and data loading, but subsequent epochs converged to runtime in the millisecond range. The training speed comparison plot reveals consistent computational efficiency across training runs, with epoch durations typically ranging from 0.001 to 0.05 seconds. This stability was maintained as the dataset increased from 15 to 790 images, demonstrating that scaling up training data does not substantially impact computational cost. This pattern demonstrates that the linear probing approach maintains computational efficiency regardless of dataset size.

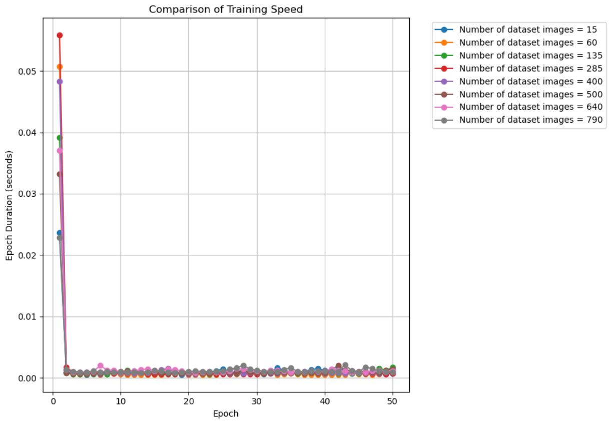

Figure 7: Training Speed Across Dataset Sizes

The result indicates the efficiency of the linear probing approach, which relies on a frozen CLIP encoder with only a lightweight trainable classification layer. The key implication is that the model can be retrained or extended quickly without requiring expensive computational resources. This consistently high training speed highlights the model's adaptability in real-world deployment. For instance, if future applications require additional gesture classes to represent more complex control commands or multimodal interactions, the system integration would only require minor retraining of the linear probe.

In the context of AcoustoBot, this efficiency directly improves the scalability of the gesture recognition interface. The interaction control system can evolve beyond the three predefined gestures (i.e., thumbs up, fist, and palm) to incorporate a broader vocabulary class, thereby supporting more functionalities or control schemes. Notably, this can be achieved without compromising training efficiency, making the approach practical for continuous development into a long-term project.

5.1.2 Training and Validation Loss

The performance of the gesture classification model is evaluated using training loss and validation loss with progressively larger dataset sizes. Training loss measures the discrepancy between the model's predicted gesture labels and the ground-truth, showing how well the system learns to map input images to the correct gesture labels during training. Training loss is computed using the cross-entropy loss function, which quantifies the difference between the predicted probability distribution (obtained after applying the softmax function to the logits) and the true encoded label distribution. Lower training loss indicates that the model is learning to correctly associate gesture images with their respective classes. Similarly, validation loss is calculated using the cross-entropy loss function but applied to the validation dataset, which is not seen during training. This metric assesses how well the model generalises unseen images. A lower validation loss reflects stronger generalisation, while a high validation loss relative to training loss may indicate overfitting. Ideally, both losses should decrease as training progresses, with the validation loss stabilising at a low value, indicating that the model is memorising the training data and learning to generalise patterns for gesture recognition.

The result is displayed in the comparison plot (Figures 8 and 9), showing that training loss and validation loss exhibit marked improvements as the dataset size increases. The training loss comparison plot demonstrates consistent convergence patterns across all runs, with training loss decreasing from initial values around 1.10 to below 0.40 by the 50th epoch. Although the dataset with only 15 images appears to converge faster than larger datasets in the training loss graph, it is a sign of severe overfitting and poor generalization. The key insight lies in examining the validation loss alongside the training loss. While the 15-image dataset shows rapid training loss reduction, its validation loss increases during the first 10 epochs (from 1.14 to 1.18) before slowly decreasing, indicating that the model memorizes the extremely small training set rather than learning generalizable features [35].

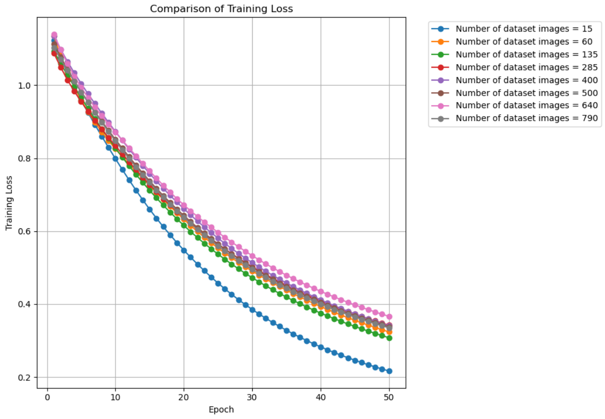

Figure 8: Training Loss Across Dataset Sizes

The reason the smaller dataset appears to converge faster is due to the fundamental principle that smaller datasets require fewer gradient updates to achieve low training loss, but this comes at the cost of poor generalization performance [5]. With only 12 training images (after the 20% validation split), the linear probe can easily memorize the exact feature representations of these samples, leading to artificially low training loss values. But in reality, the model does not learn to generalise. The larger datasets converge more slowly and have a higher training loss value because they contain more diverse and complex gesture variations, forcing the model to actually learn the underlying pattern through more training iterations and not simply memorize everything. The resulting model is significantly more robust and generalizable. This phenomenon illustrates the classic machine learning trade-off between training speed and model quality, where smaller datasets may achieve lower training loss quickly but result in models that fail to generalize to new data.

The validation loss follows a decreasing trend as the dataset increases, indicating that the model is learning meaningful representations without significant overfitting. However, some early runs show concerning patterns where validation loss initially increases while training loss decreases, suggesting potential overfitting issues that were later resolved through improved training configurations. For smaller datasets, validation loss remains relatively high and fluctuates significantly, implying that the model struggles to generalise when trained on limited data. Conversely, larger datasets yield lower and more stable validation losses, accompanied by smoother convergence in training loss. This pattern reflects the model's increased exposure to diverse gesture samples under varying conditions, which enhances its ability to generalise to real-world scenarios.

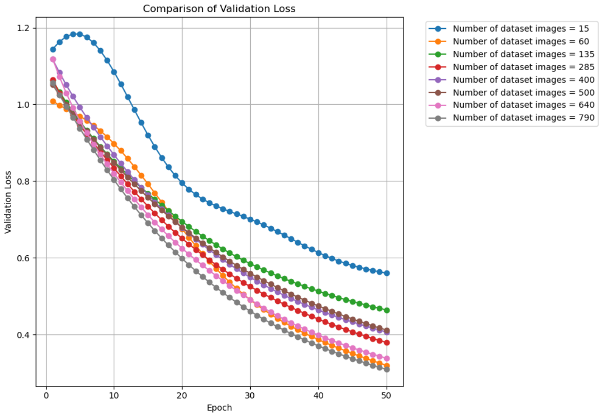

Figure 9: Validation Loss Across Dataset Sizes

The comparisons of training loss and validation loss indicate that the trained model benefits substantially from dataset scaling, reinforcing the importance of building a diverse and abundant dataset when training a gesture recognition model. The decline in validation loss establishes that the model can learn semantically meaningful gesture sets rather than overfitting to small samples. From a system design perspective, this implies that the visual learning model can reliably distinguish between the three gesture classes: thumbs up, fist, and palm, when provided with sufficiently varied training data. Furthermore, it suggests that the architecture can extend additional gestures in the future, provided that the new classes are supported with adequate training data, thereby ensuring scalability and long-term adaptability of the system.

5.1.3 Validation Accuracy

In addition to tracking the training and validation loss, the validation accuracy was recorded after each training run to evaluate the model's generalisation capability on previously unseen data. Validation accuracy, expressed as a percentage, measures the proportion of correctly classified validation samples and serves as the primary indicator of overall model performance.

The first training run, conducted with a dataset of only 15 images (five per gesture class), demonstrated limited performance and achieved a final validation accuracy of approximately 67%. As shown in Figure 10, the accuracy curve plateaued after roughly 20 epochs, suggesting that the model learned rapidly from the small dataset but could not improve further. This result highlights the model's inability to generalize effectively from the small dataset. The limited number of samples significantly reduced the diversity of visual patterns, causing the model to overfit to the small training set without obtaining an extensive un-

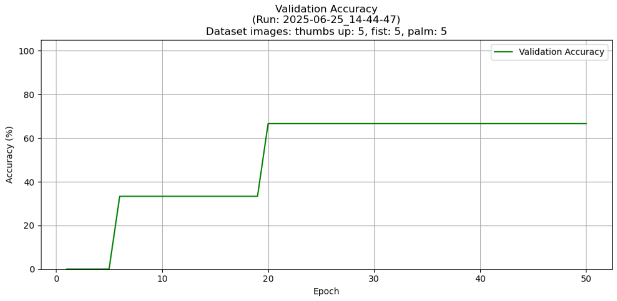

Figure 10: Validation Accuracy Curve for First Training Run (Dataset Size: 15 Images)

derstanding of the gesture classes. As a result, the model demonstrates a poor classification ability.

In contrast, the most recent training run with a substantially larger dataset comprising 790 images improved the model performance significantly, addressing the constraints observed in previous training runs with minimal data. Figure 11 illustrates an increasing trend in validation accuracy, with the model achieving stable and high classification performance close to 98%. This result demonstrates that the expanded dataset provided sufficient diversity and complexity for the model to learn robust, generalizable representations of each gesture, reducing the probability of overfitting. The high accuracy rate indicates that the system can distinguish between gesture categories with strong reliability when trained with adequate data.

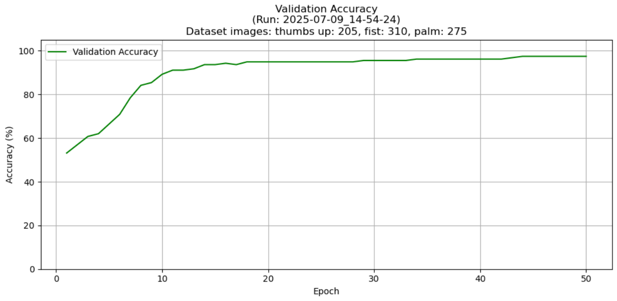

Figure 11: Validation Accuracy Curve for Latest Training Run (Dataset Size: 790 Images)

These results underscore the critical role of dataset size in reaching high recognition performance. While small datasets may allow for faster convergence, they limit the model's ability to generalize when facing unseen data. Implementing a larger, more diverse dataset enables the model to capture a richer set of gesture features, resulting in higher validation accuracy and greater reliability when deployed in real-world human-robot interaction scenarios.

5.2 Evaluation of the Integrated System with AcoustoBots

5.2.1 Experimental Setup

The integration experiment was conducted in a controlled laboratory environment (shown in Figures 12 and 13) equipped with a server PC, two AcoustoBots, two ESP32 cameras, two user handles, and a PhaseSpace tracking system. The server PC, functioning as the system's central processing and communication hub, was configured with Windows 11 (64-bit), an AMD Ryzen 7 5800H CPU, an NVIDIA GeForce RTX 3060 GPU, and 16 GB of RAM. Two AcoustoBots were equipped with acoustic control client software, each mounted with an ESP32 camera module to capture live video frames of the user's hand gestures. This configuration enabled the AcoustoBots to interpret user commands and perform multimodal actuation through gesture-based input. The PhaseSpace tracking system was employed to enable real-time motion tracking and facilitate user-following behavior. LED micro-drivers were attached to each AcoustoBot to support accurate tracking and navigation within the test arena. In addition, each user operated an LED handle to allow for precise position tracking. The positional data of both the user and the robots was transmitted to the server, which computed navigation commands to maintain user-following behavior while incorporating a collision avoidance mechanism.

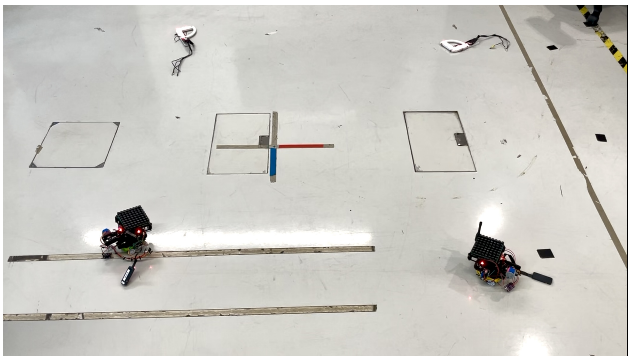

Figure 12: AcoustoBots Experiment Test Arena

Collectively, the experiment setup integrated the system's three key data streams—AcoustoBot and user position from PhaseSpace, gesture input from ESP32 cameras, and robot actuation feedback—into a coherent framework for demonstrating real-time human-swarm robot interaction.

5.2.2 Experimental Result

Experiments were conducted on the developed VLM-integrated AcoustoBot platform to evaluate the accuracy of gesture-to-modality mapping (expressed as a percentage) and the response latency (measured in

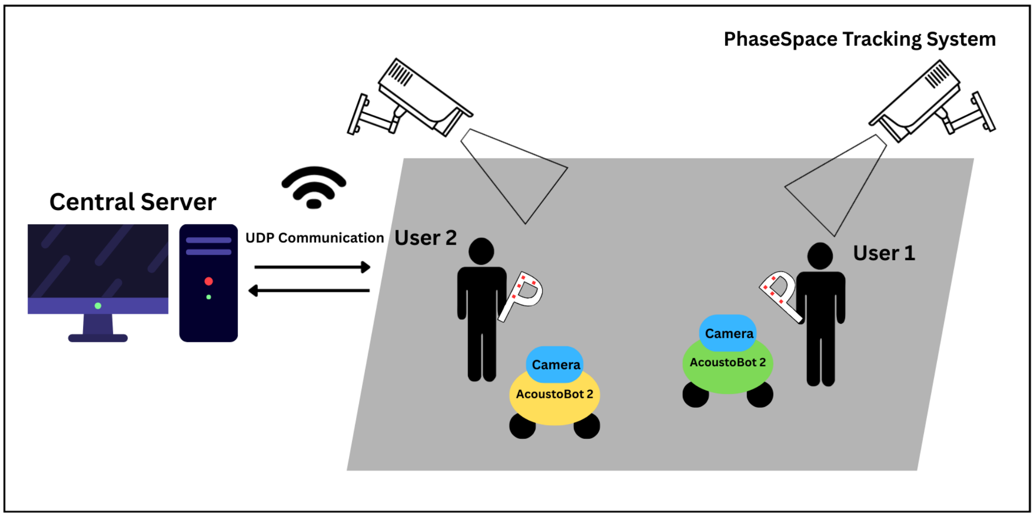

Figure 13: Experimental Setup Diagram for the Gesture Visual Learning Model System Evaluation

milliseconds) of robot actuation. Additional experiments were performed to assess the ability of the AcoustoBots to follow users in real time using the PhaseSpace tracking system.

Accuracy of Modality Switching

The accuracy of the modality switch was evaluated based on the percentage of correctly classified gesture inputs successfully translated into the corresponding AcoustoBot modality commands. The three modalities of the AcoustoBot—acoustic levitation (thumbs up), audio (fist), and haptic (palm)—were tested through multiple trials conducted in the test arena. A total of 30 trials were performed for each gesture under varying lighting and background conditions to assess robustness and classification accuracy. Recognition outputs were logged and compared with the intended gesture inputs to compute the overall classification accuracy. Gesture inputs were considered correctly classified if the model accurately interpreted the gesture and triggered the appropriate modality actuation command through the central server.

Table 1 summarizes the classification performance across gestures. The result demonstrates that the integration of the gesture classification system achieved high recognition performance across all three gesture classes, with an overall accuracy of approximately 86.7%. When broken down by class, the recognition rates were 83.3% for thumbs up, 86.7% for fist, and 96.7% for palm.

While all three gestures were recognized with relatively high accuracy, the thumbs up gesture showed lower performance than the other two categories. One contributing factor is that the thumbs up gesture is inherently more complex in its visual structure. Unlike a closed fist or open palm, which present clear, distinguishable silhouettes, the Thumbs Up gesture involves more refined details, making it more sensitive to variations in hand orientation, background, and lighting conditions. In addition, the ESP32-CAM module, which captures the input frames, has relatively low resolution and limited imaging quality. This hardware constraint makes it difficult for the system to identify finer details, such as the separation of the thumb from the rest of the hand. As a result, the thumbs up gesture is more challenging for the system to recognize consistently.

Despite these limitations, the system's overall accuracy remains robust and reliable, proving that the developed visual learning model is capable of delivering adequate performance for real-time human–robot interaction through gesture recognition. This outcome indicates that the gesture vocabulary, though small, provides an efficient interface for multimodal robot control, with room for future improvements through higher-quality sensing hardware or dataset augmentation.

Response Latency

The second evaluation metric focuses on response latency, which is defined as the elapsed time between a user performing a gesture and the corresponding AcoustoBot executing the command. Latency was recorded by embedding timestamps at each stage of the data flow, allowing precise calculation of the elapsed time between input capture and robot actuation. This method ensured that the measurements captured the true end-to-end performance of the system. Latency is a critical factor in interactive systems, as excessive delays can significantly affect user experience and reduce the system's practicality in real-world applications.

Table 1: Accuracy of modality switching across gesture classes

Gesture Class	Intended Commands (Trials)	Correctly Classified	Accuracy (%)
Thumbs Up	30	24	83.3
Fist	30	26	86.7
Palm	30	29	96.7
Overall	90	79	87.8

Latency was measured across ten trials for each modality, with the total response time decomposed into three stages: image capture and transmission (from the ESP32 camera to the server), gesture classification (server-side VLM inference), and command transmission (from the server to the AcoustoBot microprocessor). Average latencies for each stage were calculated and are presented in Figure 14. The results indicate that the majority of the delay occurs during the image capture and transmission stage, which is primarily attributed to the limited hardware capabilities of the ESP32 camera module and the high computational demand of transmitting image frames over Wi-Fi. In contrast, VLM inference time remains relatively low due to the efficiency of the linear probe design. Command transmission to the AcoustoBot contributes a moderate delay, reflecting the communication and control response characteristics of the robot hardware.

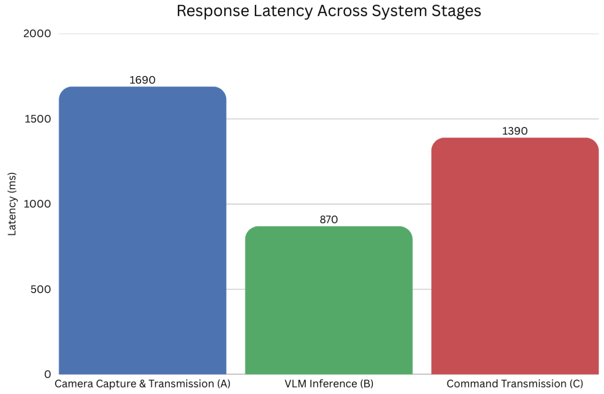

Figure 14: Average Latency Across System Stages: (A) Image capture and transmission from ESP32 Cam to Central Sever (B) Gesture classification VLM process time (C) Modality command transmission time

In addition, Figure 15 plots the distribution of response times, showing that most commands are executed in under 5 seconds. The results indicate that the system maintains an average latency of 3.95 seconds with a standard deviation of ±0.43 seconds. These performances show that the system can operate real-time human-swarm interaction, as the robots can respond to user gestures almost immediately, preserving the intuitive and interactive nature of the control interface. This latency is well below the human perceptual threshold for real-time feedback, ensuring a seamless user experience over the AcoustoBots' multimodal behaviors.

Furthermore, the analysis highlights potential areas for future optimization. Reducing camera-related latency through higher-resolution and faster imaging hardware would further improve responsiveness and scalability, enabling the system to support more complex swarm behaviors and larger user gesture vocabularies.

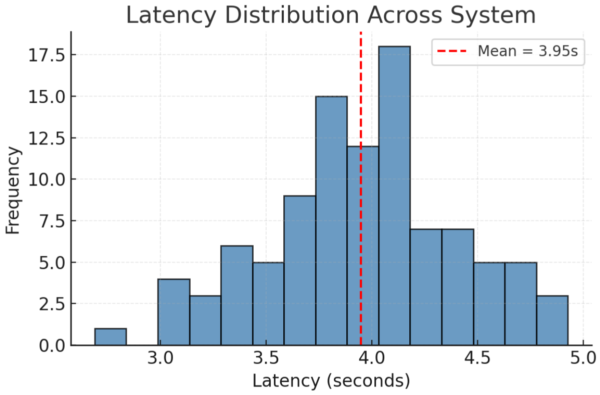

Figure 15: Distribution of Command Execution Latency Across 50 Trials

User Following Behavior

Experiments were also conducted using the PhaseSpace tracking system to evaluate the user-following behavior of the AcoustoBot system in simulated real-world human-robot interaction scenarios. The objective of this experiment was to assess how effectively the robots could follow and interact with users in dynamic environments, where responsiveness to human movement is essential. In this setup, each user held a handle marker tracked by the PhaseSpace system, enabling a one-to-one correspondence between each user and their assigned AcoustoBot. As users moved within the test arena, the robots followed them while executing modality-specific behaviors. For example, robots in haptic mode provided tactile interaction, those in audio mode emitted directional sound signals for notifications or alerts, and those in levitation mode demonstrated the ability to manipulate small particles. This experiment demonstrates the feasibility of direct human-swarm interaction, where multiple robots can simultaneously track and respond to multiple users. A demonstration of this behavior is shown in the video submission.

Currently, the hardware constraints of the PAT board limit levitation capabilities to only small lightweight particles, restricting the scope of physical object delivery. However, this experiment provides a proof-of-concept that can be extended in future work. By upgrading the PAT board and scaling up AcoustoBot numbers, the system could support larger object levitation and transport, opening opportunities for advanced applications such as autonomous delivery of small items to users, or multimodal assistance in daily environments. This potential demonstrates how the user following behavior, combined with multimodal interaction, could evolve toward practical real-world deployments.

6 DISCUSSION

This dissertation aims to address the challenge of creating a natural and intuitive interface for real-time human interaction with a swarm of multimodal robots. Specifically, the research explored the feasibility of using camera-based gesture recognition powered by a vision-language model (OpenCLIP) to control AcoustoBots—a multi-agent system capable of delivering audio, haptic, and levitation feedback through acoustic waves. To achieve this, the system integrates the ESP32-CAM module on each AcoustoBot to capture live video frames of user hand gestures. Each recognized gesture is directly mapped to a specific AcoustoBot modality: an open palm activates haptic feedback, a fist triggers audio output, and a thumbs up initiates acoustic levitation. This gesture-to-modality mapping forms the core interaction layer, allowing users to switch between functions seamlessly through natural, contactless input.

The project proved capable of using simple hand gestures to control swarm robot behavior in real time. Across evaluation trials, the system achieved a high average classification accuracy of approximately 87.8%, with a mean latency under four seconds between gesture input and robot actuation. These outcomes represent a significant step toward an accessible and user-friendly interaction interface for multi-agent systems.

Beyond demonstrating the technical feasibility of gesture-driven multimodal swarm interaction, the findings of this project offer broader reflection on how such systems can evolve and scale in the future. The primary contribution of this work lies in bridging the gap between semantic perception and physical actuation within the swarm robotics domain. Traditional approaches in human-swarm interaction have leaned heavily on symbolic inputs, scripted instructions, or abstract control languages, which, while effective in constrained environments, limit usability for general users and severely hinder deployment in real-world settings. The project breaks away from this paradigm by proposing an interaction model that aligns closely with human non-verbal communication norms: using gestures. In doing so, it establishes a novel interaction layer that is intuitive, natural, and highly accessible, making swarm systems more inclusive and practical for a broader range of application contexts.

Notably, this work advances the vision of Ichihashi et al. [18], who emphasized the importance of embodied swarm systems that respond to human movement and presence. The results also correlate with the principles outlined by Brambilla et al. [6], in which complex swarm behaviors can emerge from relatively simple local rules when embedded with a carefully designed interaction framework. While previous studies of swarm control focused primarily on movement and spatial coordination, this system integrates multimodal interaction, including haptics, audio, and levitation, thereby adding layers of functional expressivity. This project also extends the interaction space by showing that visual semantics, when interpreted through a gesture-based interface, can effectively transform user intent to multimodal swarm actuation.

At the same time, the project highlights critical limitations and trade-offs. The system relies on centralized processing, with gesture frames sent from the ESP32-CAM to a central server for classification. This design ensures reliable performance with lightweight hardware but does not achieve full decentralization, a core principle of swarm robotics. Future work should explore distributed gesture recognition, where each robot can perform local inference using more advanced embedded hardware or compressed versions of vision-language models. Such decentralization would enhance system scalability and autonomy, bringing swarm systems closer to their theoretical ideals.

The integration of acoustic levitation also proposes exciting possibilities for future applications. Currently, the hardware constrains levitation to small lightweight particles, but the framework demonstrates how users can intuitively activate this function in real time. With more advanced phased array boards and a larger number of robots, levitation-based swarms could be scaled up to perform collaborative object manipulation and delivery. In future implementations, multiple AcoustoBots could collaborate to levitate larger items stably, deliver them to users on demand, or provide dynamic haptic and audio effects in immersive environments, through gesture-based input commands. This proof-of-concept shows how gesture-driven control can act as the foundation for levitation-enabled swarms that might one day support logistics, healthcare, or interactive technologies.

Together, the outcomes of this project illustrate both the practicality and the immense possibilities of gesture-based multimodal swarm systems. This work demonstrates that simple hand gestures can reliably control complex swarm robot behaviors, providing a scalable and intuitive interaction framework. More extensively, it points toward a future where levitation-enabled swarm robots, empowered by advanced hardware and decentralized intelligence, act as collaborative agents that operate in everyday environments. Such systems have the potential to extend human capabilities, improve efficiency in daily tasks, and enhance the overall quality of life.

7 CONCLUSION

This project presented the design and implementation of a gesture-based visual learning model for controlling multimodal swarm robots. By mapping simple hand gestures to the haptic, audio, and levitation functions of the AcoustoBot, the project demonstrated that natural, non-verbal communication can serve as an effective interface for human-swarm interaction.

The system achieved reliable performance and established a practical framework for connecting human intention with embodied swarm behaviors. While limitations such as centralized processing and constrained levitation hardware remain, this work provides the foundation for scaling to larger swarms with more advanced equipment. With further development, gesture-driven interaction could evolve Acoustophoretic swarms beyond a proof of concept, shaping the next generation of swarm robotics.

ACKNOWLEDGMENTS

I would like to express my gratitude to my supervisors, Sriram Subramanian, Narsimlu Kemsaram, and Lei Gao, for their guidance and feedback throughout this project. I would like to thank Diego Martinez Plasencia and Ryuji Hirayama for their work in acoustics, which greatly strengthened my project. I would also like to thank James Hardwick for his contributions to the AcoustoBot mechanical design, and Bonot Gautam for his assistance with electronics and technical support. Their input and support were instrumental in the successful completion of this dissertation. Finally, I would like to thank my family and friends for their encouragement and support, without which this journey would not have been possible.

References

[1] Jean-Baptiste Alayrac et al. “Flamingo: a visual language model for few-shot learning”. In: *Advances in neural information processing systems* 35 (2022), pp. 23716–23736.

[2] Marco AB Andrade, Asier Marzo, and Julio C Adamowski. “Acoustic levitation in mid-air: Recent advances, challenges, and future perspectives”. In: *Applied Physics Letters* 116.25 (2020).

[3] Junjie Bai, Fang Lu, Ke Zhang, et al. *Onnx: Open neural network exchange.* 2019.

[4] Faryal Batool et al. “ImpedanceGPT: VLM-driven Impedance Control of Swarm of Mini-drones for Intelligent Navigation in Dynamic Environment”. In: *arXiv preprint arXiv:2503.02723* (2025).

[5] Christopher M Bishop and Nasser M Nasrabadi. *Pattern recognition and machine learning.* Vol. 4. 4. Springer, 2006.

[6] Manuele Brambilla et al. “Swarm robotics: a review from the swarm engineering perspective”. In: *Swarm Intelligence* 7 (2013), pp. 1–41.

[7] John Bridle. “Training stochastic model recognition algorithms as networks can lead to maximum mutual information estimation of parameters”. In: *Advances in neural information processing systems* 2 (1989).

[8] Ting Chen et al. “A simple framework for contrastive learning of visual representations”. In: *International conference on machine learning.* PmLR. 2020, pp. 1597–1607.

[9] Mehdi Cherti et al. “Reproducible scaling laws for contrastive language-image learning”. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition.* 2023, pp. 2818–2829.

[10] Damien Dablain et al. “Understanding CNN fragility when learning with imbalanced data”. In: *Machine Learning* 113.7 (2024), pp. 4785–4810.

[11] Jia Deng et al. “Imagenet: A large-scale hierarchical image database”. In: *2009 IEEE conference on computer vision and pattern recognition.* Ieee. 2009, pp. 248–255.

[12] Alexey Dosovitskiy et al. “An image is worth 16x16 words: Transformers for image recognition at scale”. In: *arXiv preprint arXiv:2010.11929* (2020).

[13] Daniele Foresti et al. “Acoustophoretic contactless transport and handling of matter in air”. In: *Proceedings of the National Academy of Sciences* 110.31 (2013), pp. 12549–12554.

[14] Ralph W Gerchberg. "A practical algorithm for the determination of plane from image and diffraction pictures". In: *Optik* 35.2 (1972), pp. 237–246.

[15] Ian Goodfellow et al. *Deep learning.* Vol. 1. 2. MIT press Cambridge, 2016.

[16] Heiko Hamann. *Swarm robotics: A formal approach.* Vol. 221. Springer, 2018.

[17] Kaiming He et al. "Deep residual learning for image recognition". In: *Proceedings of the IEEE conference on computer vision and pattern recognition.* 2016, pp. 770–778.

[18] Sosuke Ichihashi et al. "Swarm body: Embodied swarm robots". In: *Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems.* 2024, pp. 1–19.

[19] Narsimlu Kemsaram et al. "AcoustoBots: A swarm of robots for acoustophoretic multimodal interactions". In: *Frontiers in Robotics and AI* 12 (2025), p. 1537101.

[20] Nitish Shirish Keskar et al. "On large-batch training for deep learning: Generalization gap and sharp minima". In: *arXiv preprint arXiv:1609.04836* (2016).

[21] Lawrence H Kim et al. "User-defined swarm robot control". In: *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems.* 2020, pp. 1–13.

[22] Louis Vessot King. "On the acoustic radiation pressure on spheres". In: *Proceedings of the Royal Society of London. Series A-Mathematical and Physical Sciences* 147.861 (1934), pp. 212–240.

[23] James Kirkpatrick et al. "Overcoming catastrophic forgetting in neural networks". In: *Proceedings of the national academy of sciences* 114.13 (2017), pp. 3521–3526.

[24] Andreas Kolling et al. "Human interaction with robot swarms: A survey". In: *IEEE Transactions on Human-Machine Systems* 46.1 (2015), pp. 9–26.

[25] Ananya Kumar et al. "Fine-tuning can distort pretrained features and underperform out-of-distribution". In: *arXiv preprint arXiv:2202.10054* (2022).

[26] Junnan Li et al. "Align before fuse: Vision and language representation learning with momentum distillation". In: *Advances in neural information processing systems* 34 (2021), pp. 9694–9705.

[27] Hongyi Liu and Lihui Wang. "Gesture recognition for human-robot collaboration: A review". In: *International Journal of Industrial Ergonomics* 68 (2018), pp. 355–367.

[28] Ilya Loshchilov and Frank Hutter. "Decoupled weight decay regularization". In: *arXiv preprint arXiv:1711.05101* (2017).

[29] Branislav Malobický et al. "Towards Seamless Human–Robot Interaction: Integrating Computer Vision for Tool Handover and Gesture-Based Control". In: *Applied Sciences* 15.7 (2025), p. 3575.

[30] Asier Marzo, Adrian Barnes, and Bruce W Drinkwater. "TinyLev: A multi-emitter single-axis acoustic levitator". In: *Review of Scientific Instruments* 88.8 (2017).

[31] Kai Melde et al. "Holograms for acoustics". In: *Nature* 537.7621 (2016), pp. 518–522.

[32] Munir Oudah, Ali Al-Naji, and Javaan Chahl. "Hand gesture recognition based on computer vision: a review of techniques". In: *journal of Imaging* 6.8 (2020), p. 73.

[33] Diego Martinez Plasencia et al. "GS-PAT: high-speed multi-point sound-fields for phased arrays of transducers". In: *ACM Transactions on Graphics (TOG)* 39.4 (2020), pp. 138–1.

[34] Jing Qi et al. "Computer vision-based hand gesture recognition for human-robot interaction: a review". In: *Complex & Intelligent Systems* 10.1 (2024), pp. 1581–1606.

[35] Fengchun Qiao, Long Zhao, and Xi Peng. "Learning to learn single domain generalization". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition.* 2020, pp. 12556–12565.

[36] Alec Radford et al. "Learning transferable visual models from natural language supervision". In: *International conference on machine learning.* PmLR. 2021, pp. 8748–8763.

[37] Jia Chuan A Tan et al. "A proposed set of communicative gestures for human robot interaction and an RGB image-based gesture recognizer implemented in ROS". In: *arXiv preprint arXiv:2109.09908* (2021).

[38] Hung-Yu Tseng et al. "Cross-domain few-shot classification via learned feature-wise transformation". In: *arXiv preprint arXiv:2001.08735* (2020).

[39] Wei Wang et al. "A survey of zero-shot learning: Settings, methods, and applications". In: *ACM Transactions on Intelligent Systems and Technology (TIST)* 10.2 (2019), pp. 1–37.

[40] Malaika Zafar et al. “SwarmVLM: VLM-Guided Impedance Control for Autonomous Navigation of Heterogeneous Robots in Dynamic Warehousing”. In: *arXiv preprint arXiv:2508.07814* (2025).

[41] Chiyuan Zhang et al. “Understanding deep learning requires rethinking generalization”. In: *arXiv preprint arXiv:1611.03530* (2016).

[42] Kaiyang Zhou et al. “Learning to prompt for vision-language models”. In: *International Journal of Computer Vision* 130.9 (2022), pp. 2337–2348.