Image generated by DALLE-3 and modified by finalists using MacOS Preview, 2025.

THETA: Triangulated Hand-state Estimation for Teleoperation and Automation in Robotic Hand Control

ROBO052T
ALEX HUANG
AKSHAY KARTHIK
ASU SCENE
TEMPE, AZ, USA
ISEF 2025

INTRODUCTION

BACKGROUND

The global teleoperated robotics market is **projected to grow from $40.17 billion to $171.91 billion by 2032**, driven by rising automation demands across several fields **[1]**. **Teleoperation, the remote control of robots,** enables safe task execution in hazardous, inaccessible, or precision-critical areas, such as medical procedures, industrial operations, agricultural monitoring, or those with disabilities.

However, high costs, complexity, and limited accessibility of teleoperation technology restrict widespread adoption, highlighting the need for affordable solutions to enhance robotic integration.

RESEARCH GAP

- Current robotic teleoperation techniques rely heavily on costly **infrared depth cameras** and **sensor-embedded gloves**.
- **Depth cameras** like Intel RealSense D455 ($350), Microsoft Azure Kinect ($400), and high-end motion-sensor systems such as Vicon (~$10,000+) significantly raise costs, limiting accessibility for the everyday user.
  - **Google MediaPipe**, a vision-based prediction system for joint angles using trigonometry and vector math, loses accuracy when the hand is curled, perpendicular, or flexed due to landmark occlusion.
- **Finger-tracking sensor gloves** like Manus Prime X ($5,000+) and CyberGlove II ($10,000+) **[2]** further increase expenses and complexity.

Existing methods lack a cost-effective, vision-based alternative capable of accurately estimating joint angles in real-time without expensive hardware or occlusion-prone hand tracking systems.

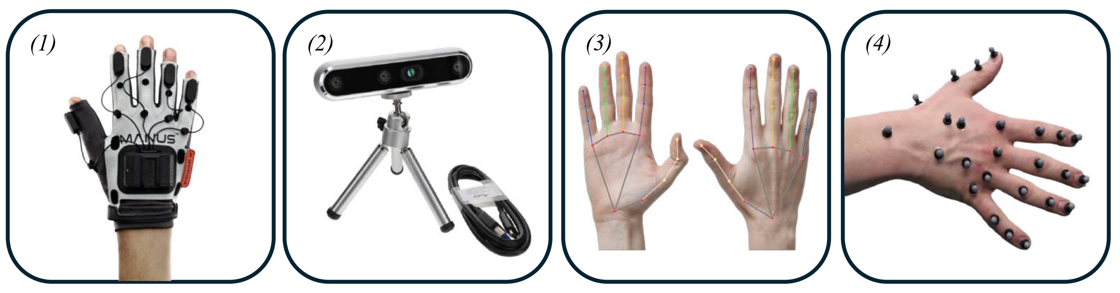

*Figure 1: Existing teleoperation technologies: (1) Manus Prime X glove ($5000+) **[3]**, (2) Intel RealSense D455 camera ($350+) **[4]**, (3) Google MediaPipe (free software, single-camera based) **[5]**, (4) Vicon system (~$10,000+) **[6]**.*

OBJECTIVES & PROPOSED SOLUTION

We present **THETA [7], a novel**, **cost-effective** method utilizing three **triangulated** inexpensive **webcams ($15 each)** for multi-view tracking to **estimate** relative **joint angles (θ)** in human fingers.

- Our approach integrates **DeepLabV3** for precise **hand segmentation** and **MobileNetV2** for robust **joint angle classification**, trained on a manually annotated dataset to enhance accuracy.
- These predictions are seamlessly transmitted to an **Arduino-controlled, low-cost (~$250), and open-sourced robotic hand**, enabling real-time, precise joint movement replication and significantly reducing system costs and complexity while maintaining **accurate, responsive** teleoperation.

NOVELTY & ADVANCEMENTS

- **Introduced THETA, a novel joint angle estimation system invariant to hand orientation, using triangulated multi-view webcam RGB input.**
  - THETA eliminates the need for hands to remain parallel to a front-facing camera, overcoming the limitations of existing landmark-based and joint-location tracking methods for joint-angle recognition.
  - THETA is at a fraction of the cost (~$45) compared to traditional depth cameras and embedded-sensor gloves ($400+).
- **Restructured & optimized CNN** (MobileNetV2) layers to enhance joint angle detection rather than generic image classification tasks.
- **Engineered a low-cost, dexterous robotic hand (~$250)** to validate the effectiveness of THETA, setting a new benchmark for affordability and adaptability in teleoperation.

EXPERIMENTAL DESIGN

1	Robotic Hand Development & ROS2 Control	Built a **servo-driven DexHand robotic hand** with modified hardware using 3D-printed parts, servos, and springs. **Developed ROS 2 software pipeline** for joint control and Arduino serial communication for hand actuation.
2	Multi-View Data Collection, Annotation & Segmentation	Collected **synchronized images of hand gestures** from multiple webcam angles; applied DeepLabV3 segmentation (ResNet-50 backbone) to isolate hands; **manually measured and annotated finger joint angles for supervised learning.**
3	Segmentation Preprocessing & THETA Joint Angle Classification	**Preprocessed segmented images** and trained a lightweight, efficient **MobileNetV2-based classifier** to accurately predict finger joint angles, optimizing performance using advanced deep-learning methods.
4	THETA Joint Angle Prediction & Real-Time Inference	**Evaluated THETA model performance**, achieving high accuracy and strong potential for generalization across diverse conditions. **Implemented real-time inference w/ serial communication** for robotic hand actuation.

METHODOLOGY

1 DEXHAND ROBOTIC HAND DESIGN, ASSEMBLY, & ROS2 CONTROL INTEGRATION

*Figure 2: Assembled DexHand (with personal modifications). **Image taken by finalists, 2025.***

- Constructed 3D Hand from DexHand CAD model and wrist mechanism by **The RobotStudio [8].**
  - Hand comprised entirely of **3D-prints**, **fishing line**, **bearings**, **springs**, **mini servos, and screws.**
- Phalanges, knuckle joints, and metacarpal bones **fastened w/ 80-lb fishing line** + 2mm spring.
- 3x Emax ES3352 12.4g mini servos (4.8-6V) and 1 spring **actuates each finger.**
  - 2x servos for **abduction/adduction** and finger base **flexion**, 1x servo for **fingertip flexion.**
- 1 spring for **fingertip** (distal) and **base** (proximal) **extension.**

- **Ubuntu VM w/ USB passthrough** used as ROS2 environment.
- Arduino pipeline relied on **two main ROS2 nodes** to facilitate robotic hand movement **[9].**
  - **Gesture Controller Node**: Generates and manages high-level hand joint angles by publishing an array of 15 servo angles on the *dexhand_hw_command* Python topic.
    - Developed new function to modify servo angles dynamically.
  - **USB Serial Node**: Acts as a **bridge** between **ROS 2** and the **Arduino**, converting high-level commands into **serial messages** that the Arduino Mega **interprets** to control the **servos** on the robotic hand.

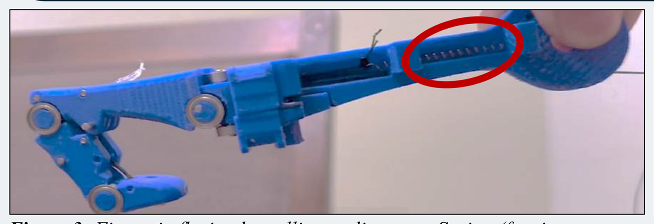

*Figure 3: Fingertip flexion by pulling on ligament. Spring (for tip extension) circled. **Image taken by finalists, 2025.***

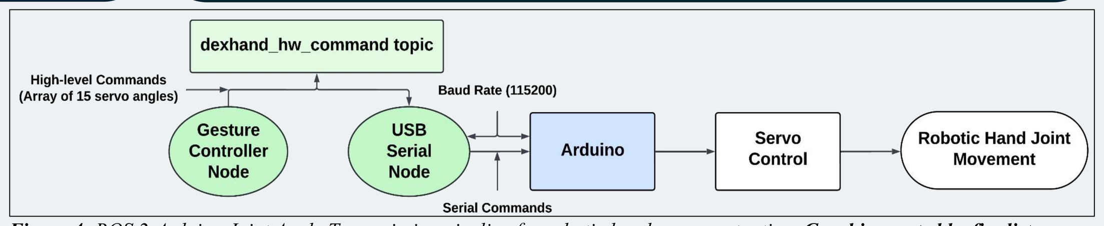

*Figure 4: ROS 2-Arduino Joint Angle Transmission pipeline for robotic hand servos actuation. **Graphic created by finalists using Lucidchart, 2025.***

2 THETA ARCHITECTURAL PIPELINE: MULTI-VIEW DATA COLLECTION, ANNOTATION & SEGMENTATION

2A STANDARDIZED GESTURE DATASET: DEFINITION, JOINT ANGLE MAPPING, DATABASE INTEGRATION

Gesture Id	Gesture Name	Index MCP Angle	Index PIP Angle	Index DIP Angle	Middle MCP Angle
1	Closed Fist	90 (±5°)	90 (±5°)	110 (±5°)	90 (±5°)
2	Open Palm	180 (±5°)	180 (±5°)	180 (±5°)	180 (±5°)
3	Number One	180 (±5°)	180 (±5°)	180 (±5°)	90 (±5°)

*Figure 5: Example entries from the "gesture joint angles" dataset, which defines 40 standardized hand gestures and maps their corresponding 15 joint angles (MCP, PIP, DIP) for each finger. **Data table created by finalists using Microsoft PowerPoint, 2025.***

A "Ground Truth gesture joint angles" dataset was manually created by measuring 15 joint angles across 40 distinct hand gestures using a protractor.

The angles of the three finger joints in each finger were recorded:

- **Metacarpophalangeal (MCP) joint**: flexion/extension, abduction/adduction at the knuckle.
- **Proximal Interphalangeal (PIP) joint:** mid-finger bending.
- **Distal Interphalangeal (DIP) joint:** fingertip actuation.

2B MULTI-VIEW RGB DATA COLLECTION FOR HAND TRACKING

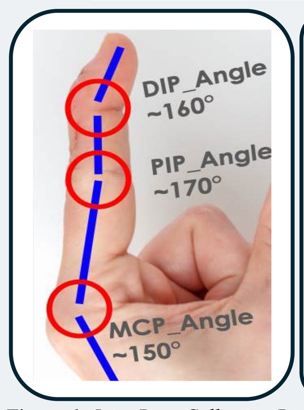

*Figure 6: Joint Data Collection Diagram [10]. **Annotations created by finalists, 2025.***

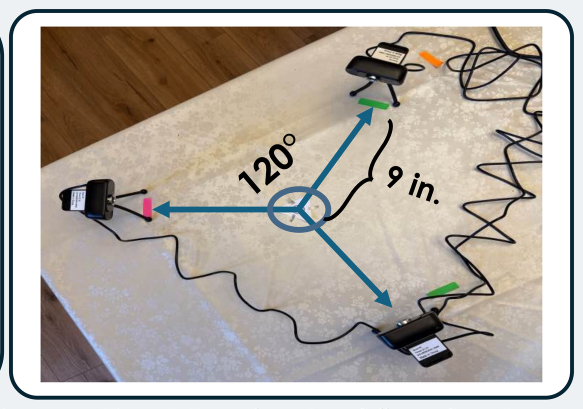

*Figure 7: Triangulation Data Collection Setup. **Image and annotations taken by finalists, 2025.***

Synchronized RGB images (640×480, 30 FPS) are captured from three webcam angles, front, right, and left, while performing the selected hand gesture. The corresponding joint angles are recorded, with a ±5-degree perturbation applied per frame to enhance variability and improve generalization.

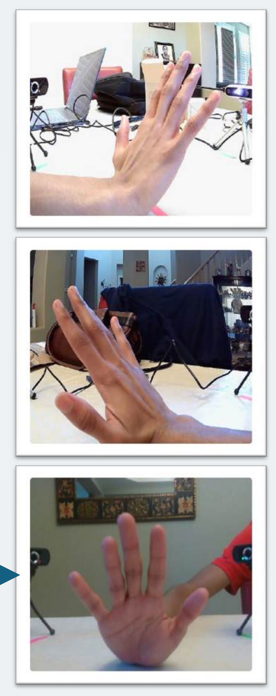

*Figure 8: Synchronized RGB images captured from multiple viewpoints (right, left, and front). **Images taken by finalists, 2025.***

2C HAND SEGMENTATION: DATA PROCESSING PIPELINE & MASK GENERATION

Image Preprocessing → **Feature Extraction** → **Segmentation Prediction** → **Post-Processing & Refinement**

Image Preprocessing

- Resize images to 224×224 for model input

$$I_{\text{resized}} \in \mathbb{R}^{224\times224\times3}$$

*Figure 9: Resized RGB image tensor shape. **Graphic created by finalists using Lucidchart, 2025.***

- Normalize pixel values

$$\mu = \begin{bmatrix} 0.485 & 0.485 & \dots & 0.485 \\ 0.456 & 0.456 & \dots & 0.456 \\ 0.406 & 0.406 & \dots & 0.406 \end{bmatrix} \quad \sigma = \begin{bmatrix} 0.229 & 0.229 & \dots & 0.229 \\ 0.224 & 0.224 & \dots & 0.224 \\ 0.225 & 0.225 & \dots & 0.225 \end{bmatrix}$$

*Figure 10: Image normalization mean (μ) and standard deviation (σ) values. **Graphic created by finalists using Lucidchart, 2025.***

- Convert images to tensors

Feature Extraction

- Pass images through DeepLabV3 with ResNet-50 backbone **[11].**
- Apply Atrous Spatial Pyramid Pooling for multi-scale feature extraction

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

*Figure 11: Feature extraction adjacency matrix: Nodes represent feature extraction stages, and edges (1s) denote transformations or feature extraction steps. **Graphic created by finalists using Lucidchart, 2025.***

Segmentation Prediction

- Optimize masks using BCEWithLogitsLoss
- Track accuracy using Mean IoU
- Convert soft masks into binary segmentation masks
- Apply erosion/dilation for noise removal and mask refinement
- Red overlay mask is applied to the segmented hand region

Post-Processing & Refinement

- Resize masks using nearest-neighbor interpolation
- Convert final segmentation masks to binary format
- Extract segmented hand regions for THETA model input

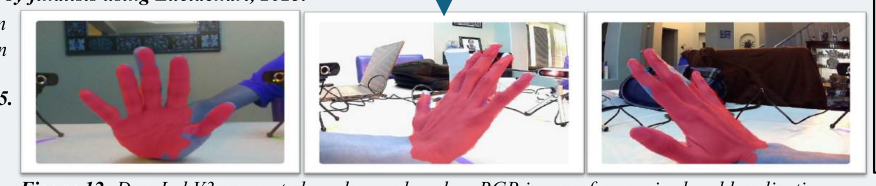

*Figure 12: DeepLabV3-generated masks overlayed on RGB images for precise hand localization across different perspectives. **Images created by finalists, 2025.***

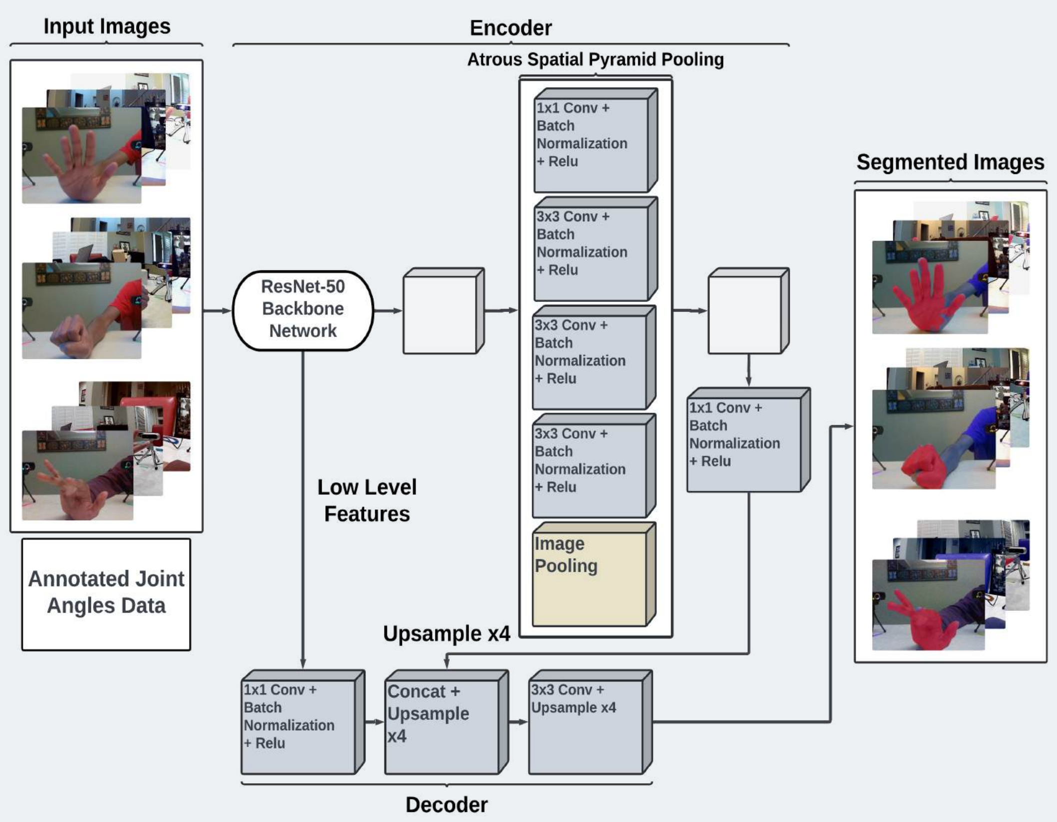

*Figure 13: THETA End-to-End Segmentation Pipeline: Multi-View RGB Image Segmentation w/ DeepLabV3 for Image Preprocessing, Feature Extraction, Segmentation Prediction, and Mask Generation. **Graphic created by finalists using Lucidchart, 2025.***

3 THETA ARCHITECTURAL PIPELINE: SEGMENTATION PREPROCESSING & JOINT ANGLE CLASSIFICATION

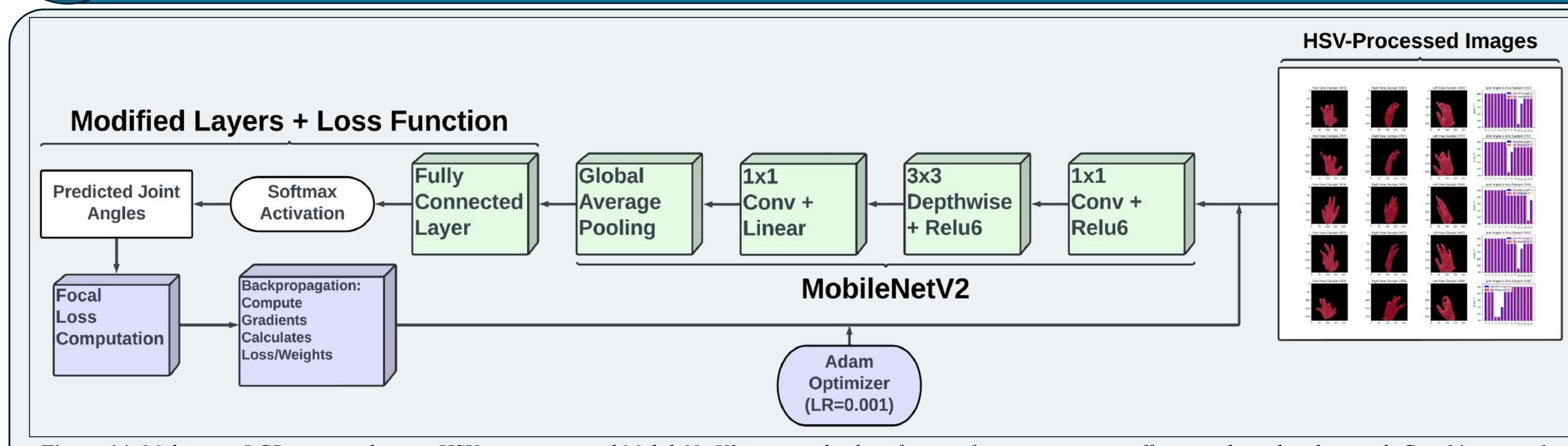

*Figure 14: Multi-view RGB input undergoes HSV processing and MobileNetV2 joint angle classification for accurate, cost-effective robotic hand control. **Graphic created by finalists using Lucidchart, 2025.***

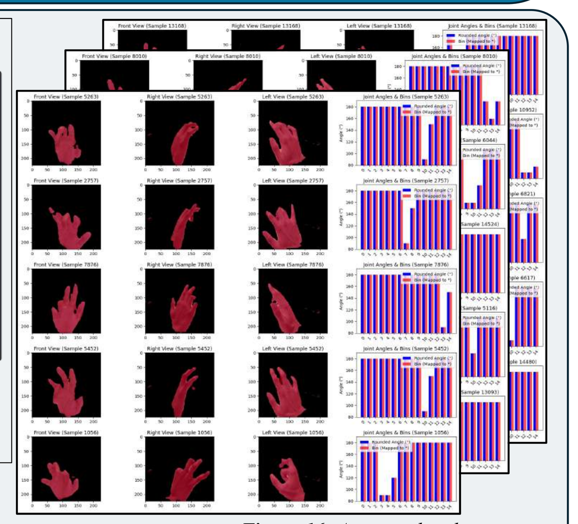

*Figure 16: Accurate hand segmentation across triangulated views. **Images and graphs created by finalists using Matplotlib, Seaborn, and OpenCV, 2025.***

- **HSV Segmentation:** Isolates hands via red color thresholds.
- **Data Processing:** Normalizes, resizes (224×224), and bins joint angles.
- **Feature Extraction:** MobileNetV2 classifies joint angles from multi-view images.
- **Optimization:** Focal Loss (γ=2.0), Adam (LR=0.001), 10-epoch training.
- **Evaluation:** Predicts joint angles, tracks accuracy, refines probabilities.
- **Modified Last Layer:** Reshapes to (batch, 15, 10), applies T=2.0 scaling, and softmax.

$$\begin{bmatrix} H & S & V \\ 0-10 & 120-255 & 70-255 \\ 170-180 & 120-255 & 70-255 \end{bmatrix}$$

*Figure 15: HSV color thresholding matrices for hand segmentation. The predefined lower and upper red thresholds effectively isolate hand regions from the background. **Graphic created by finalists using Lucidchart, 2025.***

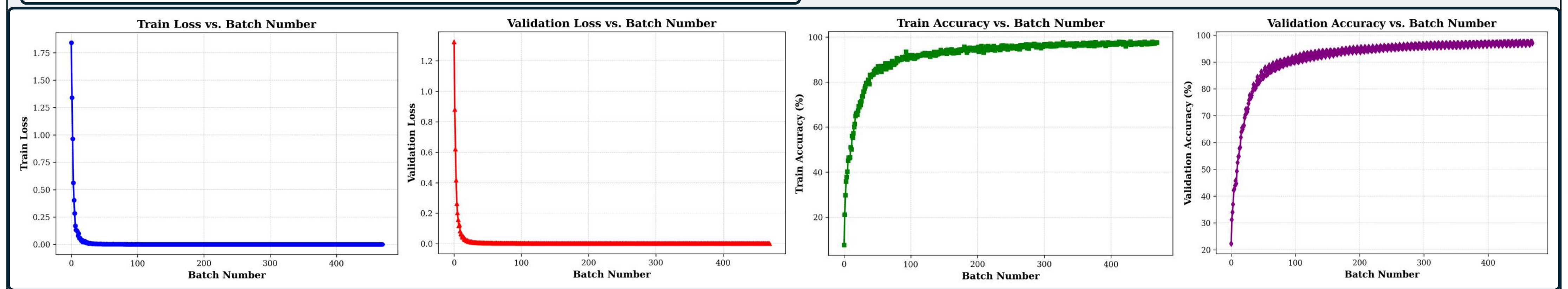

*Figure 17: 97.50% training accuracy and 97.03% validation accuracy with loss convergence to 0.0001, demonstrating strong generalization, minimal overfitting, and reliable joint angle classification for real-time robotic hand teleoperation. **Graphs created by finalists using Matplotlib and Seaborn, 2025.***

CONCLUSION

4 JOINT ANGLE PREDICTION & INFERENCE

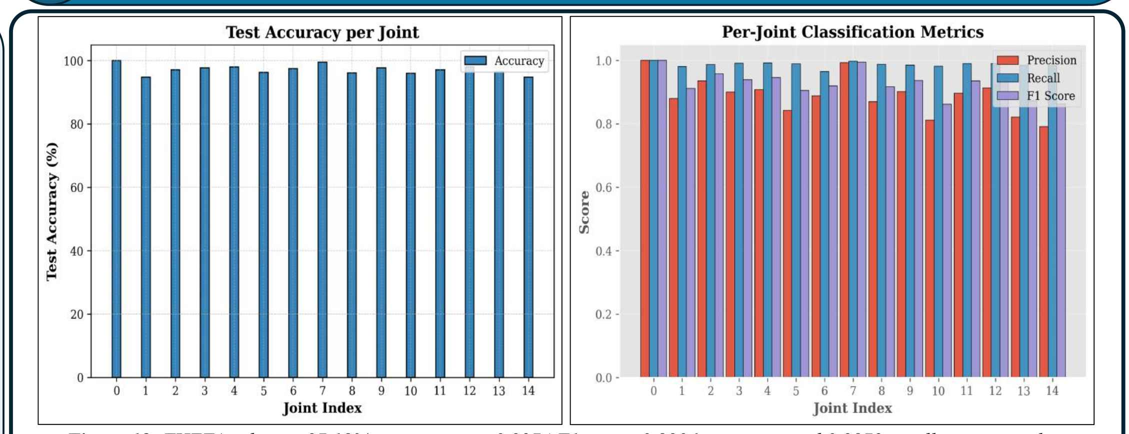

*Figure 18: THETA achieves 97.18% test accuracy, 0.9274 F1-score, 0.8906 precision, and 0.9872 recall in joint angle classification, ensuring precise hand-state estimation. **Graphs created by finalists using Matplotlib and Seaborn, 2025.***

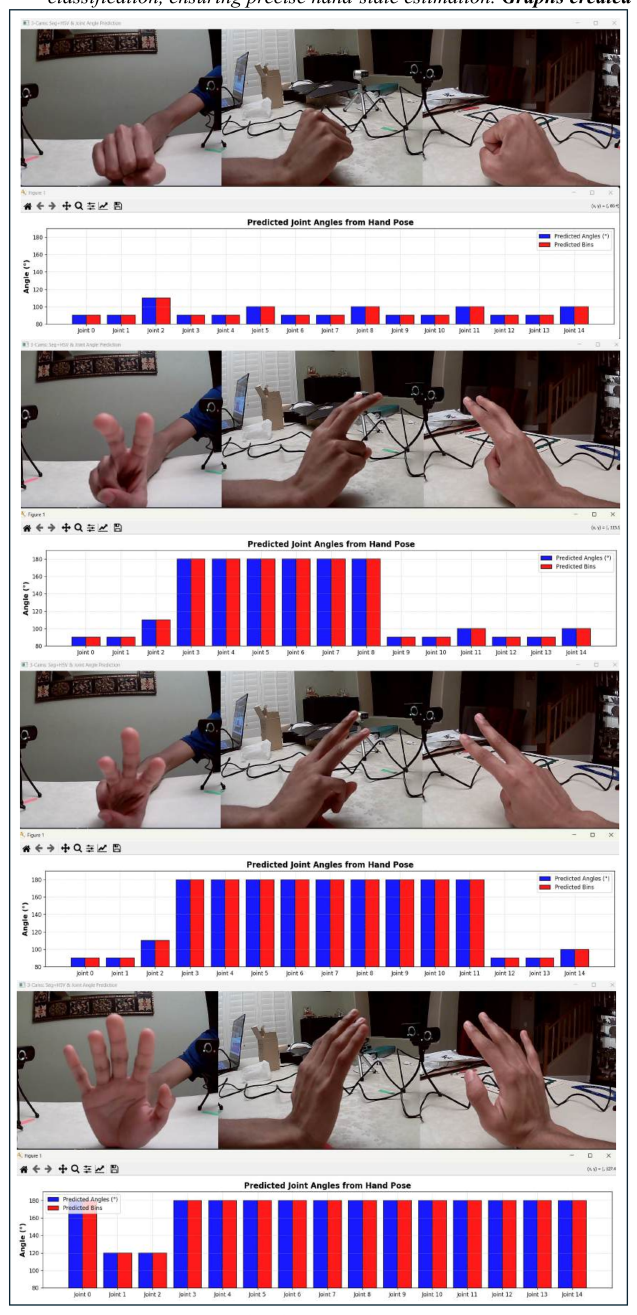

*Figure 19: Real-time joint angle inference pipeline using THETA's multi-view triangulation for precise and responsive robotic hand control. **Images taken and graphs created by finalists using Matplotlib, Seaborn, and OpenCV, 2025.***

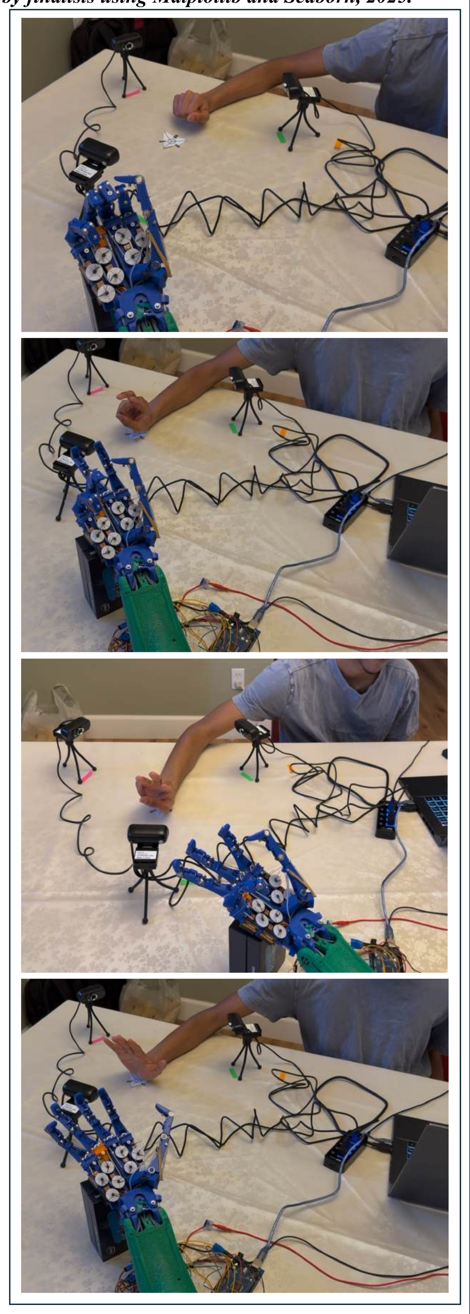

*Figure 20: THETA real-time joint angle prediction and inference with serial communication to the DexHand using Arduino. **Images taken by finalists, 2025.***

RESULT EVALUATION

- THETA achieved **97.18% accuracy on the testing set**, demonstrating **high potential for strong generalization in predicting joint angles**.
  - The model attained an F1 score of **0.9274**, precision of **0.8906**, and recall of **0.9872**, ensuring **precise hand pose estimation** for robust motion analysis.
- In our project video, the THETA-DexHand pipeline **successfully mimicked triangulated joint angles**, validating real-world applicability.

THETA's simple setup and robustness has the potential to increase the accessibility of high-compliant teleoperated robotic hands, with implications for countless real-life fields.

LIMITATIONS

- **Despite having over 48,000 training images**, THETA's data sample size remains limited due to the limited and costly nature of training and computation on cloud GPUs.
  - **As the dataset size increases, THETA can transition from a classification-based approach to a regression model,** enabling more precise and accurate continuous joint-angle predictions.

FUTURE RESEARCH & APPLICATIONS

Develop adaptive learning models that continuously refine and enhance joint angle recognition through **weighted user feedback.**

Integrate LLM reasoning, logic, and **image capabilities** to enhance compliance and awareness for situational contexts.

Graphic created by finalists using Microsoft PowerPoint, 2025.

Optimize deep learning pipelines to minimize latency and boost real-time responsiveness of physical robotic hand.

APPLICATIONS

Household Prosthetics: Improve automation and AI functionalities in household prosthetics, especially for those with disabilities.

Medical Field: Support remote surgical procedures with advanced, precise joint angle recognition technology.

Linguistics: Facilitate remote or automated sign language interpretation and gestures, such as American Sign Language.

Space Exploration: Enable the manipulation of extraterrestrial objects during space missions, like material sampling on exoplanets.

Graphic created by finalists using Microsoft PowerPoint, 2025.

REFERENCES

1.Robotics Market Size, Growth Trends, Report and Forecast 2022-2027. (n.d.). Www.imarcgroup.com. https://www.imarcgroup.com/robotics-market
2.Avian, C., Prakosa, S. W., Faisal, M., & Leu, J.-S. (2022). Estimating finger joint angles on surface EMG using Manifold Learning and Long Short-Term Memory with Attention mechanism. Biomedical Signal Processing and Control, 71, 103099. https://doi.org/10.1016/j.bspc.2021.103099
3.MANUS. (2024). MANUS Prime X Glove. Website-Files.com. https://assets-global.website-files.com/61f1346be9b2882b854a2ece/6332b1ced5f966ea15a5fe4a_Prime%20Series%20Gloves%20by%20MANUS-p-800.png
4.DigiKey. (2025). 82635DSD456 | DigiKey Electronics. DigiKey Electronics. https://www.digikey.com/en/products/detail/intel-realsense/82635DSD456/21555839
5.APMonitor. (2023). Hand Tracking. Apmonitor.com. https://apmonitor.com/pds/index.php/Main/HandTracking
6.Le, H. V., Mayer, S., Steuerlein, B., & Henze, N. (2019). Investigating Unintended Inputs for One-Handed Touch Interaction Beyond the Touchscreen. 1–14. https://doi.org/10.1145/3338286.3340145
7.smokyfishy. (2025). GitHub - smokyfishy/THETA. GitHub. https://github.com/smokyfishy/THETA
8.Open, A. (2023). DexHand - An Open Source Dexterous Humanoid Robot Hand. DexHand - an Open Source Dexterous Humanoid Robot Hand. https://www.dexhand.org
9.iotdesignshop. (2023). GitHub - iotdesignshop/dexhand_ros2_meta: Metapackage for DexHand ROS 2 Packages. Used to manage the collection of ROS 2 packages and environment for the DexHand humanoid robot hand. GitHub. https://github.com/iotdesignshop/dexhand_ros2_meta
10.Mallet Finger - Charleston Hand, Wrist, and Elbow - Hand Specialists - Charleston, SC. (2023, December 8). Charleston Hand, Wrist, and Elbow - Hand Specialists - Charleston, SC. https://charlestonhandspecialist.com/hand-specialist/mallet-finger/
11.guglielmocamporese. "GitHub - Guglielmocamporese/Hands-Segmentation-Pytorch: A Repo for Training and Finetuning Models for Hands Segmentation." GitHub, 4 Aug. 2022, www.github.com/guglielmocamporese/hands-segmentation-pytorch/tree/master.