REAL-TIME MONOCULAR 3D HAND DISTENCE ESTIMATION USING A QUADRATIC REGRESSION MODEL

DO HUU SON[1], PHAN VAN NAM[2], TRAN HUU PHUC [2*]
[1]*Ha Noi University of Science and Technology, Ha Noi, Vietnam*
[2]*Post and Telecommunications Institute of Technology, Ha Noi, Vietnam*
Email address: phuctranhuu37@gmail.com
**Corresponding author: Tran Huu Phuc*
http://doi.org/10.51453/3093-3706/2026/1468

ARTICLE INFO

Received: 09/02/2026
Revised: 12/3/2026
Published: 28/4/2026

KEYWORDS

3D Hand;
3D Distance Estimation;
Deep Learning.

ABSTRACT

This research addresses the problem of estimating the distance from a user's hand to a conventional 2D monocular camera in a 3D space, aiming to optimize low-cost Human-Machine Interfaces (HMIs) and virtual reality interactions. In contrast to expensive dedicated depth sensors such as lasers, ultrasound, or RGB-D cameras, the proposed method integrates the MediaPipe Hands framework to detect 21 skeletal hand landmarks in real-time, and applies a quadratic nonlinear regression model to calculate the physical distance. To overcome geometric distortions caused by hand closure/opening gestures or wrist rotations, the pixel distance between two geometrically stable landmark pairs—namely landmarks 5–17 (representing the horizontal axis) and landmarks 9–0 (representing the vertical axis)—is extracted as the independent input variables. The final estimated distance is refined through a minimum filter to suppress tracking noise.

Empirical evaluations on a standard consumer laptop demonstrate that the model operates highly efficiently in real-time with minimal resource overhead, consuming only 176,6 MB of RAM and 20,5% of CPU. The model is evaluated across three distinct illumination environments (well-lit, dimly-lit, and outdoor) within a physical distance range of 22 cm to 117 cm. Experimental results show that the proposed model achieves outstanding accuracy at interaction distances between 50 cm and 100 cm, yielding a minimum relative error of only 0,02% at 90 cm. A rigorous comparative study with a standard linear regression model validates the superiority of the quadratic nonlinear approach in modeling the perspective-scaling characteristics of 2D monocular projection.

1. INTRODUCTION

1.1. Study Motivation

In the current era of digital transformation, the rapid evolution of information technology has generated an urgent demand for optimizing Human-Machine Interfaces (HMIs), especially the capability to accurately track and localize targets in a 3D coordinate space.

Presently, 3D tracking and distance measurement rely heavily on active sensors such as ultrasonic transceivers, LiDAR, or depth-sensing cameras (e.g., Intel RealSense). However, these active systems suffer from high implementation costs, complex calibration procedures, and limited

compatibility with standard consumer electronic devices. Consequently, there is significant interest in leveraging a standard 2D monocular camera to infer depth and estimate 3D spatial distance. This approach offers several compelling advantages:

* Exceedingly low cost, as it can be deployed immediately on any camera-equipped consumer device (webcams, smartphones, standard laptops).

* Seamless integration with existing video surveillance or front-facing camera infrastructures without requiring hardware modifications.

* High potential for real-world applications in smart home control, touchless interactive kiosks, robot navigation, and virtual/augmented reality (VR/AR).

Despite these advantages, extracting absolute 3D depth from 2D monocular perspective images presents notable mathematical and computational challenges. The inferred pixel coordinates are highly sensitive to fluctuating illumination, motion blur, varied camera viewing angles, and the severe non-rigid deformation of human hands during natural gesturing. In this context, this study presents a robust framework titled Research on Distance Estimation using a 2D Camera to develop an accurate, resource-friendly, and highly applicable hand distance estimation pipeline.

1.2. Reseach Objectives and scope

Objective: Build an optimized quadratic regression model that maps 2D projected hand landmark coordinates to absolute physical distances (cm) with minimal error, and compare it against a linear baseline.

Scope: Real-time 2D hand landmark extraction and absolute depth (cm) estimation under various environments.

1.3. Methodology

Modeling: Construct a quadratic regression model to represent the inverse relationship between 2D pixel scale and 3D physical distance.

Implementation: Code the pipeline in Python using OpenCV, MediaPipe, and NumPy to process standard video streams in real-time.

Evaluation: Collect ground-truth distance data (using a tape measure) to compute and compare relative errors.

2. RELATED WORKS

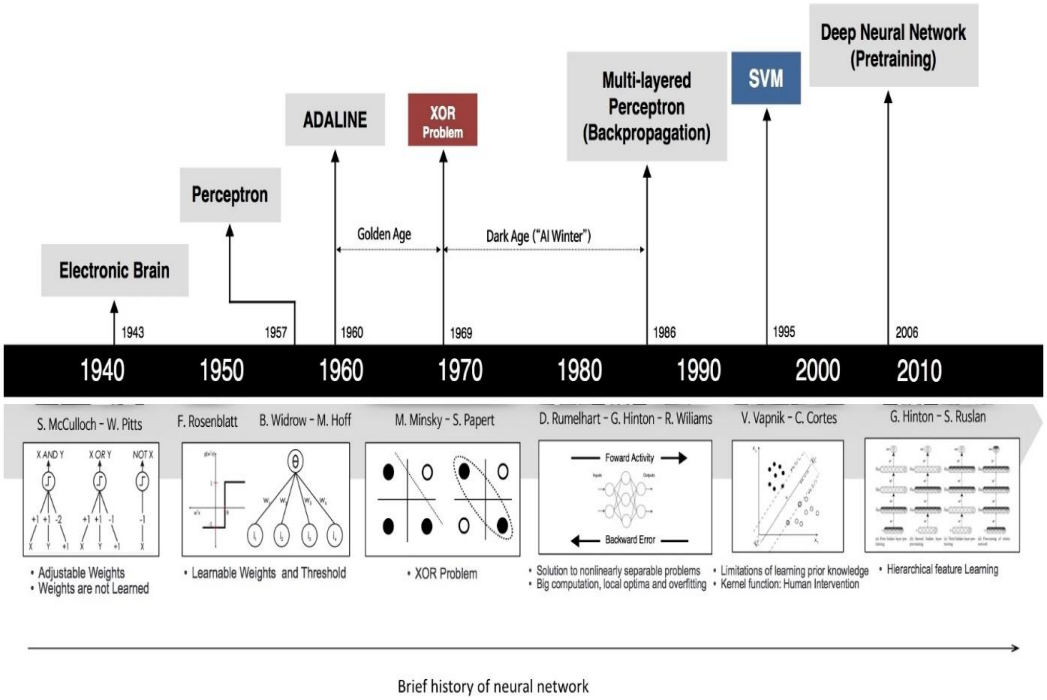


Figure 1: Timeline of Deep Learning (DL) Development

2.1. Deep Learning (DL) Foundations in Computer Vision

Deep Learning (DL) has emerged as a dominant paradigm in computer vision, utilizing multi-layered neural networks to automatically extract hierarchical feature representations directly from raw inputs [1]. To understand this paradigm shift, Fig. 1 illustrates the chronological evolution of deep learning frameworks over the past few decades, highlighting their exponential growth in complexity and capacity.

DL operates as a critical subset within the broader taxonomy of Machine Learning (ML) and Artificial Intelligence (AI). The conceptual hierarchy and overlap among these three domains are illustrated in Fig. 2, defining how each subfield inherits and extends the capabilities of its parent domain.

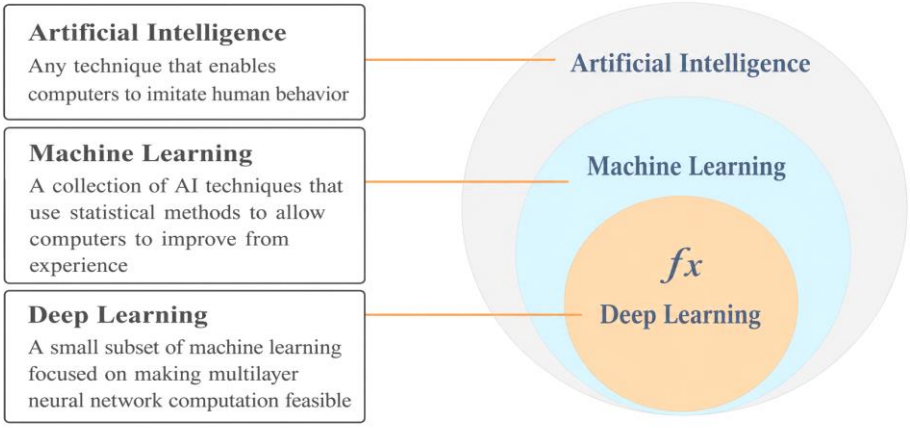


Figure 2: Conceptual Hierarchy among AI, Machine Learning (ML), and DL.

At the core of deep learning is the Artificial Neural Network (ANN), which models biological brain structures using layers of interconnected nodes. Fig. 3 provides a structural

Email: sjttu@tqu.edu.vn

representation of a typical multilayer feedforward neural network, showcasing how input data propagates forward through hidden representations to the output layer.

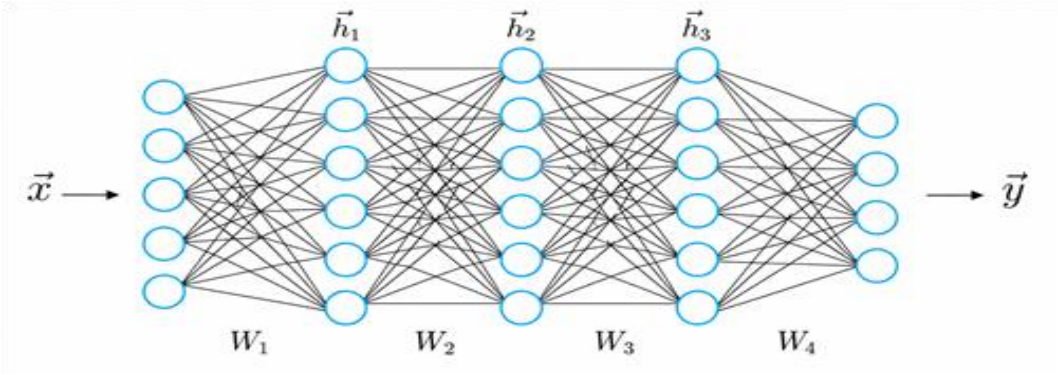


Figure 3: Topology of a Multilayer Artificial Neural Network (ANN)

2.2. International Hand Detection Studies

2.2.1. CNN-Based Hand Detection and Pose Estimation

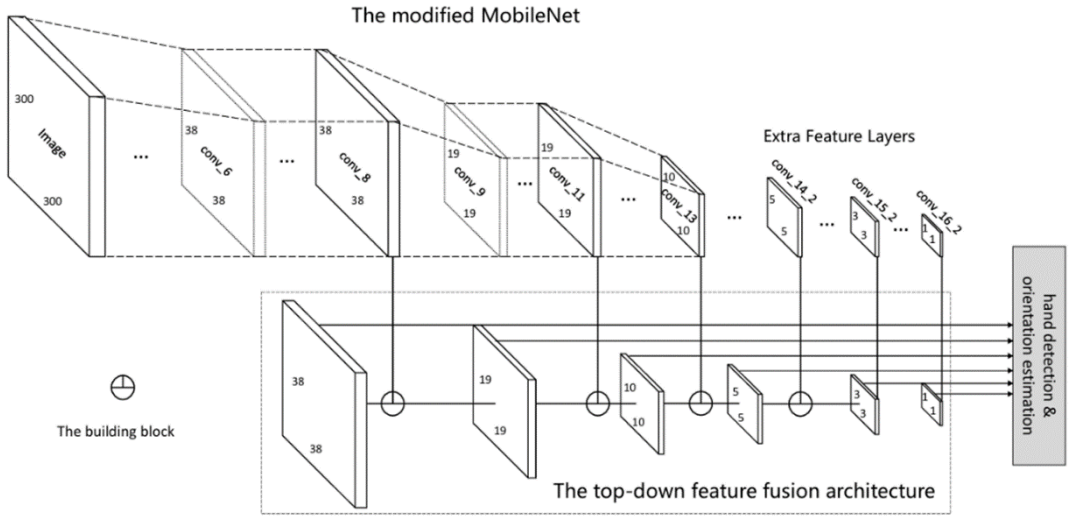


Figure 4: SSD-MobileNet Network Architecture with Top-Down Feature Fusion [3]

Li Yang et al. [3] proposed a highly efficient Convolutional Neural Network (CNN) framework designed for joint hand detection and orientation estimation. The pipeline leverages a Single Shot Multibox Detector (SSD) architectuqare backed by a customized MobileNet backbone. The overall network structure, detailed in Fig. 4, extracts six feature maps at varying resolutions to robustly detect hands across various spatial scales, utilizing a top-down feature fusion block to preserve the contextual features of smaller hands.The method achieves a highly competitive Average Precision (AP) of 83.2% on the Oxford Hand Dataset, significantly outperforming early benchmarks by Deng [5] and Le [6]. As shown in Table 1, the customized SSD-MobileNet method yields a remarkable processing time of 0.0072 seconds (139 FPS) on an Nvidia Titan X GPU, making it highly suitable for real-time edge applications.

Table 1: Performance comparison on the Oxford Hand Dataset

Method	Average Precision (AP)	Processing Time	Hardware

Mittal et al. [4]	48,20%	2,0000 min	CPU 2.5 GHz
Deng et al. [5]	57,70%	0,1000 sec	GPU Titan X
Le et al. [6]	75,10%	0,2150 sec	GPU Titan X
Proposed Method [3]	**83,20%**	**0,0072 sec**	**GPU Titan X**

2.2.2. Multiscale Convolutional Networks for Hand Detection

To address extreme scale variations and partial hand occlusions, Shiyang Yan et al. [7] integrated VGG16 with multiscale Region of Interest (RoI) pooling. The framework, illustrated in Fig. 5, integrates intermediate features from Conv3, Conv4, and Conv5 layers with L2 normalization to preserve structural details of smaller hands. The network was rigorously benchmarked on the VIVA Hand Detection Challenge and Oxford Hand Detection datasets. The corresponding AP metrics, tabulated in Table 2, demonstrate that the multiscale RoI integration achieves superior accuracy under difficult conditions, yielding 92,8% L1 AP on VIVA.

Table 2: AP comparison on the VIVA Hand Dataset

Method	L1 Set (Easy)	L2 Set (Hard)
CNNRegionSampling [8]	66,80%	57,80%
ACFDepth4 [9]	70,10%	60,10%
YOLO [10]	76,40%	69,50%
FRCNN [11]	90,70%	**86,50%**
Multiscale Fast R-CNN [7]	**92,80%**	**84,70%**

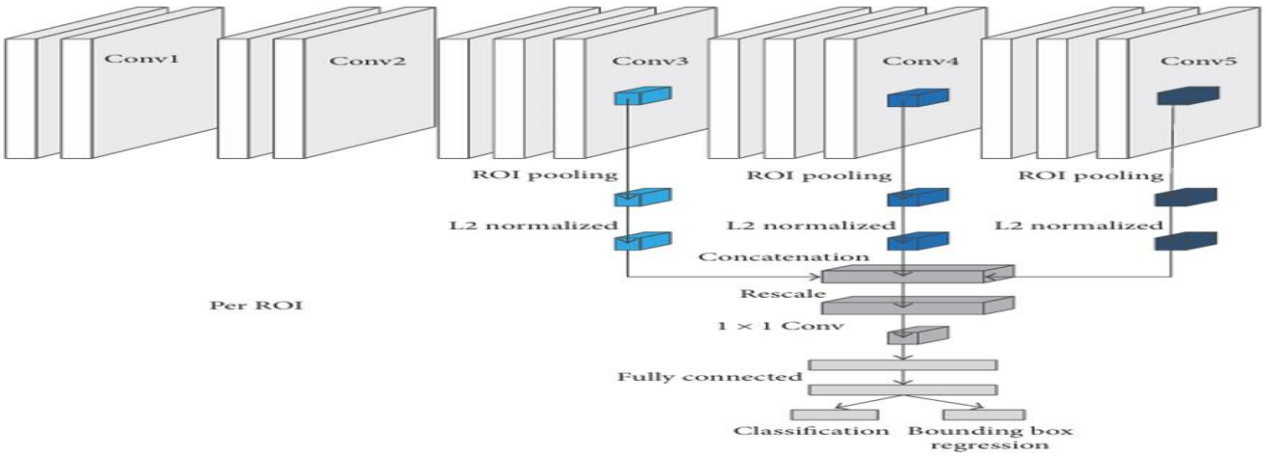


Figure 5: Detailed Multiscale Region of Interest (RoI) Pooling Structure.

2.3. Monocular and Stereo RGB Depth Estimation Methods

Depth Networks [14]: Modern methods use deep neural networks to infer depth maps from single images [15], stereo pairs [16], or video streams [17]. A representative network topology is shown in Fig. 6, consisting of three specialized sub-networks: a global network for macro structure, a gradient network for edges, and a refining network for details.

Geometric Methods [18]: Stereo camera triangulation [19] maps physical depth Z using focal length f, baseline B, and pixel disparity: $Z = \frac{f \cdot B}{disparity}$ Alternatively, Object Scaling [20] calculates depth by mathematically comparing the projected pixel size against the known physical proportions of the target.

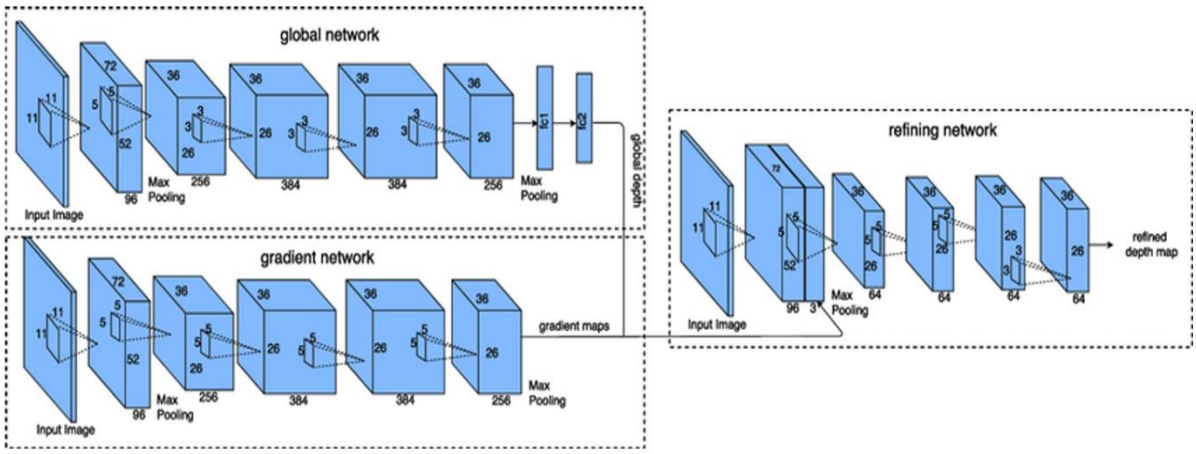


Figure 6: Global, Gradient, and Refining Sub-Network Topology for Depth Maps [14].

3. PROPOSED METHODOLOGY

3.1 Quadratic Nonlinear Regression Theory

Nonlinear regression [21] models complex physical systems by mapping a dependent variable y to independent inputs $y = f(x, \theta) + \epsilon$ Where θ represents the parameters to be estimated. To ensure statistical validity, the model must satisfy core regression assumptions: correct functional form [22], independence [23], homoscedasticity [24], normality [23], and absence of multicollinearity [25]. Algorithms are categorized into parametric [26] and non-parametric [27] approaches.

Under perspective projection, 2D projected scales change as an inverse quadratic function of physical depth. Hence, we adopt a quadratic regression model [28], formulated as: $y = \beta_0 + \beta_1 x + \beta_2 x + \epsilon$ Fig. 8 presents the various curve styles generated by the quadratic model under different parameter weights, demonstrating its suitability for mapping perspective distortion.

Email: sjttu@tqu.edu.vn

3.2. Hand Landmark Tracking via MediaPipe

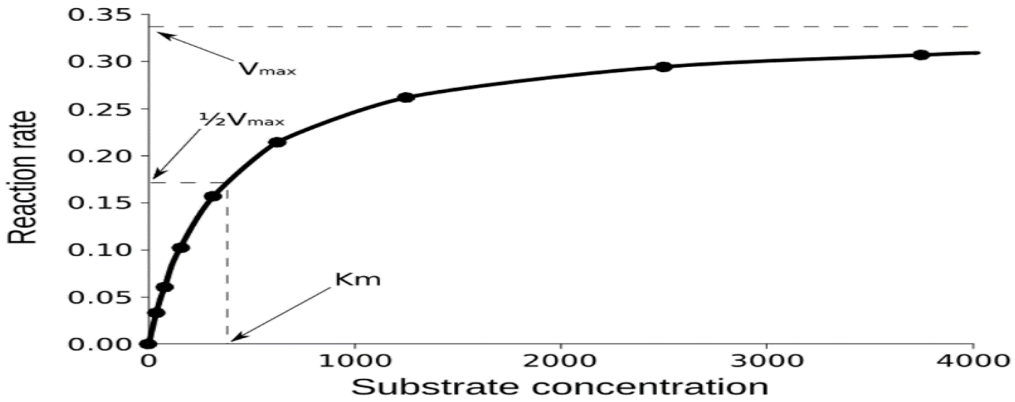


Figure 7: Illustrates a typical nonlinear regression curve fitting experimental data points, demonstrating its capacity to model curved boundaries.

We integrate the open-source MediaPipe framework [29] for high-speed, real-time hand landmark tracking. The MediaPipe Hands module [30] detects and tracks 21 skeletal landmarks on the hand, outputting their coordinates in real-time. The topological index and structural relationships of these 21 key skeletal points are shown in Fig. 9.

3.3. Proposed Distance Estimation Model

Because 2D projected scale decreases quadratically as physical distance increases, this relationship can be mapped using a parabolic curve, as illustrated in Fig. 10.

To maintain robust depth estimation when the hand curls, clenches, or rotates out-of-plane, we proposed tracking two orthogonal landmark pairs simultaneously:

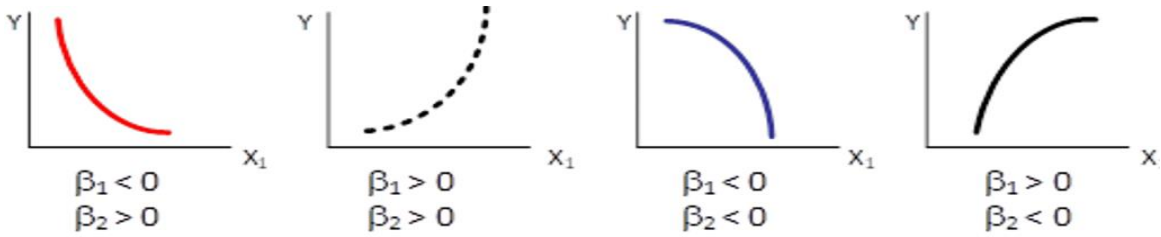


Figure 8: Characteristic Curve Profiles Associated with the Quadratic Model.

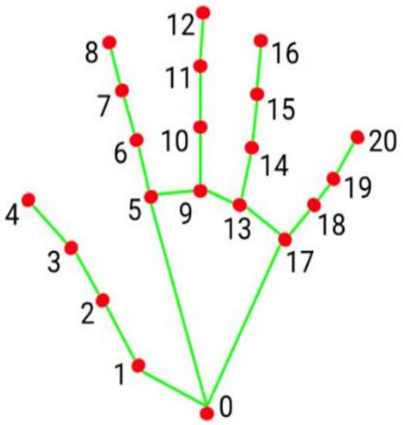
Landmarks 5 and 17: Horizontal palm span.

Landmarks 9 and 0: Vertical palm span.

This orthogonal design ensures that at least one axis remains parallel to the image plane during complex hand gestures.

Mathematical Model

The distance for each tracking line is formulated as $y = ax^2 + bx + c + \epsilon$ Where y is the



0. WRIST
1. THUMB_CMC
2. THUMB_MCP
3. THUMB_IP
4. THUMB_TIP
5. INDEX_FINGER_MCP
6. INDEX_FINGER_PIP
7. INDEX_FINGER_DIP
8. INDEX_FINGER_TIP
9. MIDDLE_FINGER_MCP
10. MIDDLE_FINGER_PIP
11. MIDDLE_FINGER_DIP
12. MIDDLE_FINGER_TIP
13. RING_FINGER_MCP
14. RING_FINGER_PIP
15. RING_FINGER_DIP
16. RING_FINGER_TIP
17. PINKY_MCP
18. PINKY_PIP
19. PINKY_DIP
20. PINKY_TIP

Figure 9: Index Mapping of the 21 Skeletal Landmarks in MediaPipe Hands.

physical distance (cm), x is the Euclidean pixel distance, and a, b, c are the regression coefficients. The pixel distance x between landmarks x_1, y_1 and x_2, y_2 is: $d = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$. NumPy's polyfit function applies the least-squares method to optimize the coefficients a, b, c over the calibration dataset. To counter tracking noise, the final physical distance D_{final} takes the minimum value:

$$D_{final} = \min(D_{5-17}, D_{9-0})$$

3.4. Experimental Setup and Calibration.

Hardware Specs: Gigabyte G5 GD (Intel i5-11400H, 16GB RAM, NVIDIA RTX 3050 Laptop GPU).

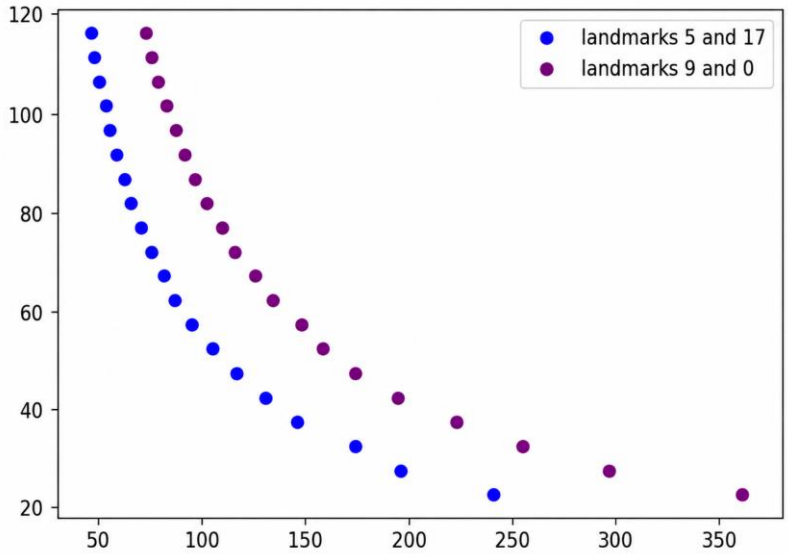


Figure 10: Empirical Parabolic Relationship Mapping Distance against Pixel Spacing.

Software Environment: Windows 11, Python 3.10.15, OpenCV 4.5.5, MediaPipe 0.10.8,

NumPy 1.26.4. Calibration Protocol: Videos were recorded at 1280x720 @ 30 FPS. The hand was positioned from 22 to 117 cm with strict 5 cm steps (tape measure verified). The experimental setup is shown in Fig. 11.

Email: sjttu@tqu.edu.vn

The resulting calibration dataset mapping the physical distances to the projected pixel spans is presented in Table 3.

Figure 11: Calibration Physical Setup for Measuring Distance vs. Pixel Spacing

Table 3: Calibration dataset of actual distance vs. projected pixel coordinates

Physical (cm)	Pixel (5 - 17)	Pixel (9 - 0)	Physical (cm)	Pixel (5 - 17)	Pixel (9 - 0)
22	240	360	**72**	75	115
27	195	295	**77**	70	109
32	174	253	**82**	65	101
37	145	222	**87**	62	96
42	130	193	**92**	58	91
47	116	173	**97**	55	87
52	105	158	**102**	53	82
57	95	148	**107**	50	78
62	86	134	**112**	48	75
67	81	126	**117**	46	72

4. RESEARCH RESULTS

4.1. Resource profile and enviroments

Resources: Real-time processing consumes 176.6 MB RAM and 20.5% CPU with high power efficiency.

Lighting Conditions: Tested on 3 illumination scenarios: Person 1 (Well-lit indoors), Person 2 (Dim light with noise), and Person 3 (Natural outdoor sunlight). Fig. 12 displays sample frames extracted from the tracking sequences under these three lighting conditions.

4.2. Proposed HQPTT Distance Estimates vs Actual Ground Truth

The physical distance predictions from our quadratic model are shown below:

Person1 Person2 Person3

Figure 12: Sample Tracking Frames in 3 Lighting Conditions (Person 1, 2, 3)

Table 4: Predicted distance values (cm) from the proposed HQPTT model

Actual (cm)	Person 1	Person 2	Person 3	Actual (cm)	Person 1	Person 2	Person 3
30	22,71	25,46	22,72	**70**	71,4	76,68	75,47
35	28	33,09	29,53	**75**	78,49	85,09	80,33
40	34,52	41,83	36,67	**80**	81,27	87,04	84,28
45	41,83	47,34	42,96	**85**	85,08	93,07	88,02
50	48,43	55,41	50,79	**90**	90,02	97,24	92,36
55	54,36	63,21	56,38	**95**	93,07	99,37	94,1
60	62,43	67,89	64,59	**100**	95,74	101,53	98,3
65	65,59	74	69,03				

- Standard (Person 1): Highly stable; relative error remains consistently under 10%.
- Dim (Person 2): Jitter in low-light slightly increases tracking error between 55–90 cm.
- Outdoor (Person 3): Shadows at close ranges (under 40 cm) introduce minor tracking offsets.

4.3. Comparison with Baseline Linear Regre (HQTT)

To evaluate performance, we trained a Linear Regression model (HQTT) on the same dataset.

Table 5: Distance predictions from the baseline linear model (cm)

Actual (cm)	Person 1	Person 2	Person 3	Actual (cm)	Person 1	Person 2	Person 3
30	28,28	35,68	27,94	**70**	77,14	80,11	79,38
35	39,71	47,71	42,07	**75**	80,6	84,06	81,4
40	48,45	54,51	51,14	**80**	82,58	85,55	83,75
45	56,19	61,9	57,53	**85**	84,56	89,52	86,04
50	62,3	67,62	62,91	**90**	87,53	90,5	88,12
55	66,26	72,19	67,95	**95**	88,52	90,99	89,01

Email: sjttu@tqu.edu.vn

60	71,7	75,01	72	**100**	89,51	92,47	90,81
65	74,67	78,13	75,35				

The table below shows the absolute relative error (%) under standard lighting (Person 1):

Table 6: Comparative absolute relative error (%) between HQPTT and HQTT models.

Actual (cm)	Quadratic (HQPTT)	Linear (HQTT)	Actual (cm)	Quadratic (HQPTT)	Linear (HQTT)
30	24,30%	**5,73%**	**70**	**2,00%**	10,20%
35	20,00%	**13,46%**	**75**	**4,65%**	7,47%
40	**13,70%**	21,13%	**80**	**1,59%**	3,22%
45	**7,04%**	24,87%	**85**	**0,09%**	0,52%
50	**3,14%**	24,60%	**90**	**0,02%**	2,74%
55	**1,16%**	20,47%	**95**	**2,03%**	6,82%
60	**4,05%**	19,50%	**100**	**4,26%**	10,49%
65	**0,91%**	14,88%			

- Close range (<35 cm): Linear regression shows lower error because landmarks often clip past the camera borders at extremely close ranges, disrupting the quadratic fit.

- Golden Interaction Range (40–100 cm): The proposed quadratic model excels, maintaining error under 5% and achieving optimal precision at 90 cm (relative error of just 0,02%) and 85 cm (0,09%). Meanwhile, the linear model peaks at a 24,87% error at 45 cm.

5. CONCLUSION AND FUTURE DEVELOPMENT

5.1. Conclusions

Developed a monocular hand distance estimation system using a quadratic least-squares regression model. Utilized two orthogonal MediaPipe landmark axes (5–17 and 9–0) to resolve hand gestural and pitch/yaw rotation errors. Achieved an optimal relative prediction error of 0.02% within standard HMI working distances. Built a lightweight, real-time Python implementation suitable for standard consumer electronics.

5.2. Future Work

Self-Supervised Auto-Calibration: Remove manual measurement steps via deep self-calibration networks.

Target Generalization: Expand to track faces, full bodies, or standardized objects via YOLO frameworks.

Deep Depth Fusion: Fuse the geometric quadratic regression with monocular depth estimation models.

HMI Implementations: Integrate into smart touchless panels, accessibility controllers, and human-robot interfaces.

REFERENCES

1. Xing Hao, Guigang Zhang, and Shang Ma, “Deep Learning,” 2016.

2. Krogh, “What are artificial neural networks?,” 2008.

3. Li Yang, Zhi Qi, Zeheng Liu, Shanshan Zhou, Yang Zhang, Hao Liu, Jianhui Wu, Longxing Shi, "A Light CNN based Method for Hand Detection and Orientation Estimation," National ASIC System Engineering Research Center, Southeast University, Nanjing, China, 2018.

4. A. Mittal, A. Zisserman, and P.H. Torr, "Hand detection using multiple proposals," 2011.

5. X. Deng, Y. Zhang, S. Yang, P. Tan, L. Chang, Y. Yuan, et al., "Joint hand detection and rotation estimation using CNN," 2017.

6. A.G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, et al., "Efficient convolutional neural networks for mobile vision applications," 2017.

7. Shiyang Yan, Yizhang Xia, Jeremy S. Smith, Wenjin Lu, and Bailing Zhang, "Multiscale Convolutional Neural Networks for Hand Detection," 2017.

8. Bambach S., Lee S., Crandall D. J., Yu C., "Lending a hand: detecting hands and recognizing activities in complex egocentric interactions," 2015.

9. Das N., Ohn-Bar E., and Trivedi M. M., "On performance evaluation of driver hand detection algorithms: challenges, dataset, and metrics Mercedes," 2015.

10. Redmon J., Divvala S., Girshick R., Farhadi A., "You only look once: Unified real-time object detection," 2016.

11. Zhou T., Pillai P. J., Yalla V. G., "Hierarchical context-aware hand detection algorithm for naturalistic driving," 2016.

12. Mittal A., Zisserman A., Torr P., "Hand detection using multiple proposals," 2011.

13. Minh-Thanh Le, Van-Ca Phan, Hai-Trang Dang Phuoc, Duy-Tan Do, Ngoc-Son Truong, "Hand Gesture Recognition Using Convolutional Network," 2021.

14. Alican Mertan, Damien Jade Duff, Gozde Unal, "Single image depth estimation: An overview," 2022.

15. A. Bhoi, "Monocular Depth Estimation: A Survey," 2019.

16. Hamid Laga, Laurent Valentin Jospin, Farid Boussaid, Mohammed Bennamoun, "A Survey on Deep Learning Techniques for Stereo-Based Depth Estimation Mercedes," 2022.

17. Onur Özyeşil, Vladislav Voroninski, Ronen Basri, Amit Singer, "A survey of structure from motion," 2017.

18. M.O. Katanaev, "Geometrical methods in mathematical physics," 2013.

19. Richard I. Hartley, Peter Sturm, "Triangulation," 2002.

20. Edward P.F. Chan, Kevin K.W. Chow, "On multi-scale display of geometric objects," 2001.

21. Gallant, "Nonlinear Regression," 2012.

22. J. S. Morris, "Functional Regression," 2015.

23. Lange, K., Sinsheimer, J. S, "Normal/Independent Distributions and Their Applications in Robust Regression," 2012.

24. Kun Yang, Justin Tu, Tian Chen, "Homoscedasticity: an overlooked critical assumption for linear regression," 2019.

25. J. I. Daoud, "Multicollinearity and Regression Analysis," 2017.

26. C. Büchel, R.J.S. Wise, C.J. Mummery, and A. Yatchew, "Nonparametric Regression Techniques in Economics," 1998.

27. J. F. Fowler, "The linear-quadratic model," 2014.

28. C. Lugaresi, J. Tang, H. Nash, C. McClanahan, E. Uboweja, M. Hays, F. Zhang, C.-L. Chang, M.G. Yong, J. Lee, W.-T. Chang, W. Hua, M. Georg, and M. Grundmann, "MediaPipe: A Framework for Building Perception Pipelines," 2019.

29. F. Zhang, V. Bazarevsky, A. Vakunov, A. Tkachenka, G. Sung, C.-L. Chang, and M. Grundmann, "MediaPipe Hands: On-device Real-time Hand Tracking," 2020.