**Gesture Recognition for Enabling Control of Electrical Devices**

by David T. Tung

B.S. in Mechanical Engineering, National Sun Yat-Sen University

Master of Computer Science, University of Southern California

A Praxis submitted to

The Faculty of

The School of Engineering and Applied Science

of The George Washington University

in partial fulfillment of the requirements

for the degree of Doctor of Engineering

May 2025

Praxis directed by

Shahryar Sarkani  
 Adjunct Professor of Engineering Management and Systems Engineering

John Fossaceca  
 Professorial Lecturer of Engineering Management and Systems Engineering

The School of Engineering and Applied Science of George Washington University certifies that David T. Tung has passed the Final Examination for the degree of Doctor of Engineering as of xx, 2025. This is the final and approved form of the Praxis.

**Gesture Recognition for Enabling Control of Electrical Devices**

David T. Tung

Praxis Research Committee:

Shahryar Sarkani, Adjunct Professor of Engineering Management and Systems Engineering, Praxis Co-Director

John Fossaceca, Professorial Lecturer of Engineering Management and Systems Engineering, Praxis Co-Director

-------, Associate Professor of Engineering and Applied Science, Committee Member [place holder]

© Copyright 2025 by David T. Tung

All rights reserved

# Dedication

I would like to dedicate this research to my family…(todo)

# Acknowledgements

I wish to thank Dr. Shahryar Sarkani and Dr. John Fossaceca for their guidance and support throughout the studying, research, and writing processes. [todo]

# Abstract of Praxis

**Gesture Recognition for Enabling Control of Electrical Devices**

Individuals with mobility impairments frequently encounter obstacles in accessing and managing electrical devices. To improve accessibility for this demographic, the implementation of a gesture-based control system should be considered. This research initiative aims to develop and assess a user-friendly, economically feasible machine-learning methodology that allows users to control devices via pointing gestures. In particular, this initiative seeks to enhance DeePoint, a three-dimensional pointing direction prediction model (Nakamura *et al.*, 2023), into a cohesive two-stage machine learning system capable of recognizing the electrical devices toward which a user is pointing. The primary objective of this project is to establish technology that can be easily developed into applications empowering mobility-impaired individuals to quickly and intuitively utilize their household devices through natural pointing gestures.

**Table of Contents**

[Dedication 4](#_Toc194739477)

[Acknowledgements 5](#_Toc194739478)

[Abstract of Praxis 6](#_Toc194739479)

[List of Figures 10](#_Toc194739480)

[List of Tables 11](#_Toc194739481)

[List of Symbols 12](#_Toc194739482)

[List of Acronyms 13](#_Toc194739483)

[Chapter 1—Introduction 1](#_Toc194739484)

[1.1 Background 1](#_Toc194739485)

[1.2 Research Motivation 3](#_Toc194739486)

[1.3 Problem Statement 4](#_Toc194739487)

[1.4 Thesis Statement 4](#_Toc194739488)

[1.5 Research Objectives 5](#_Toc194739489)

[1.6 Research Questions and Hypotheses 5](#_Toc194739490)

[1.7 Scope of Research 6](#_Toc194739491)

[1.8 Research Limitations 6](#_Toc194739492)

[1.9 Organization of Praxis 7](#_Toc194739493)

[Chapter 2—Literature Review 9](#_Toc194739494)

[2.1 Introduction 9](#_Toc194739495)

[2.2 Assistive Living and Technologies Review 10](#_Toc194739496)

[2.3 Pointing Gesture Recognition and Object Interaction 12](#_Toc194739497)

[2.4 Vision Transformer Technology 18](#_Toc194739498)

[2.5 Object Detection Technology 22](#_Toc194739499)

[2.6 Summary and Conclusion 26](#_Toc194739500)

[Chapter 3—Methodology 28](#_Toc194739501)

[3.1 Introduction 28](#_Toc194739502)

[3.2 Collect data 30](#_Toc194739503)

[3.2.1 Configure AWS Infrastructure for Data Storage and Processing 30](#_Toc194739504)

[3.2.2 Datasets for Model Training 31](#_Toc194739505)

[3.3 Preprocess Data 34](#_Toc194739506)

[3.4 Perform pointing direction estimation (Stage-1) 35](#_Toc194739507)

[3.4.1 Stage-1 Architecture 35](#_Toc194739508)

[3.5 Perform device detection & localization (Stage-2) 36](#_Toc194739509)

[3.6 Train model to classify pointing-device alignment (Stage-3) 38](#_Toc194739510)

[3.6.1. Stage-3 Model Architecture 38](#_Toc194739511)

[3.6.2 Training data 39](#_Toc194739512)

[3.6.3 Model Training 40](#_Toc194739513)

[3.7 Evaluate how gaze direction impacts accuracy 43](#_Toc194739514)

[3.7.1 Incorporate the gaze direction feature in Stage-3 44](#_Toc194739515)

[3.4.1 Modify the lost function to include the gaze direction feature 44](#_Toc194739516)

[3.4.2 Re-Train Stage-3 44](#_Toc194739517)

[3.8 Assess the impact of model choice on Stage-3 outcomes 44](#_Toc194739518)

[3.8.1 Setup system 45](#_Toc194739519)

[3.8.2 Run Experiment with MLP Model Architecture 45](#_Toc194739520)

[3.8.3 Compare Performance with the Baseline Model 45](#_Toc194739521)

[3.9 Build a single pipeline connecting all stages 45](#_Toc194739522)

[3.9.1. End-to-End Architecture 45](#_Toc194739523)

[3.9.2 Setup system 46](#_Toc194739524)

[3.10 Evaluate the Performance of the End-to-End pipeline 46](#_Toc194739525)

[3.11 Summary 47](#_Toc194739526)

[Chapter 4—Results 49](#_Toc194739527)

[4.1 Introduction 49](#_Toc194739528)

[4.2 Another Section 49](#_Toc194739529)

[Chapter 5—Discussion and Conclusions 50](#_Toc194739530)

[5.1 Discussion 50](#_Toc194739531)

[5.2 Conclusions 50](#_Toc194739532)

[5.3 Contributions to Body of Knowledge 50](#_Toc194739533)

[5.4 Recommendations for Future Research 50](#_Toc194739534)

[References 51](#_Toc194739535)

[Appendix A 61](#_Toc194739536)

# List of Figures

**No table of contents entries found.**

# List of Tables

[3.2.3 Data Visualization 33](#_Toc194739565)

[3.5.1 Object Detection Model 36](#_Toc194739566)

[3.5.2. Depth Estimation Model 37](#_Toc194739567)

[3.5.3 3D Localization Module 37](#_Toc194739568)

[3.6.3.1 Transformer Encoder and Multi-head Self-attention 41](#_Toc194739569)

[3.6.3.2 Learning Rate Scheduler 42](#_Toc194739570)

[3.6.3.3 Loss Function 42](#_Toc194739571)

[3.6.3.4 Outputs 42](#_Toc194739572)

# List of Symbols

State of the system

Output of the system

Noise

# List of Acronyms

ADA Americans with Disabilities Act

HUD Housing and Urban Development.

YOLO You Only Look Once is a real-time object detection system

CNN Convolutional Neural Networks

Fast R-CNN Fast Region-Based Convolutional Neural Networks

DL Deep Learning

HRI Human-Robot Interaction

HOI Human-Object Interaction

ViT Vision Transformer

CLIP Contrastive Language-Image Pretraining

MGM Multimodal Guidance Module

NLP Natural Language Processing

ML Machine Learning

SLAM Simultaneous Localization and Mapping

HHS Health and Human Services

CMU Carnegie Mellon University

ArUco Augmented Reality University of Cordoba

DP DeePoint

# Chapter 1—Introduction

## 1.1 Background

As the number of older adult households continues to grow, more homeowners are expected to undertake expensive accessibility renovations (U.S. Department of Housing and Urban Development, n.d., 2015). One of the most critical and costly elements of these projects is improving access to electrical devices, particularly for individuals with mobility impairments who rely on wheelchairs for movement. In addition, the number of relevant caregivers is shrinking as the need for care rapidly increases (Nora S. *et al*., 2020). Despite advances in assistive devices, laws, and technology aimed at aiding daily activities, people with mobility disabilities continue to face barriers in utilizing electrical devices, reducing their capacity to live independently.

Accessible housing design features can significantly increase the independence of people with mobility-related disabilities within their homes. Over the years, various solutions have been developed to assist individuals with mobility impairments in controlling household devices, including voice-activated systems and augmented or mixed-reality smart glasses (Zhou K. *et al*., 2023). Despite their potential, these technologies are not yet ready to be used widely within this group due to several challenges. Voice-activated systems, for instance, often underperform in noisy environments, making them unreliable in busy or public settings. They are additionally unsuitable for environments where quiet is required, such as a baby’s room or a shared living space. While smart glasses provide a hands-free solution, they also come with drawbacks, such as high costs and discomfort when worn for extended periods, which limits their practicality for long-term, everyday use.

As for the use of gestures, many existing gesture-based control systems rely on specialized hardware and sensors, which are often effective but also expensive, bulky, and cumbersome to use and maintain. Several associated research initiatives are: “AR Smart Home: A Smart Appliance Controller Using Augmented Reality Technology and a Gesture Recognizer” (Inomata *et al*., 2020), which requires an AR device, and “Magic Ring: A self-contained gesture input device on the finger” (Jing *et al*., 2013), which requires the user to wear a specially designed ring. These limitations prevent widespread adoption, particularly for individuals seeking affordable and convenient solutions to increase their independence. As a result, there is a growing need for more user-friendly, intuitive, and cost-effective approaches to overcome these barriers and provide greater accessibility in daily device control (Chang *et al*., 2022).

Creating an intuitive system that enables device control through pointing gestures using a cost-effective, fixed-mounted camera could greatly assist those with mobility challenges. For instance, corner-mounted cameras would allow a wheelchair user to move around a room and switch on lights using gestures—an installation that is likely both simple and inexpensive. Our praxis introduces a machine learning system that serves as the core feature for device control, identifying the device that the user points at. The configuration is tailored for environments containing indoor objects and has been tested with simulated data for a seated user’s pointing gesture. The machine learning system processes video feeds that track the user’s movements, detect their intent to operate devices by pointing at them and potentially generate corresponding commands for the devices to execute[[1]](#footnote-1).

With our proposed machine learning system, we can establish a foundation for future touchless device controls in other environments like elevator buttons, pedestrian push buttons, or parking lot push buttons (see Figure 1). Going forward, our project aims to provide machine learning tools to develop a more accessible and independent world.

A yellow button on a pole

Description automatically generated

Figure 1. Potential application using pointing gesture

## 1.2 Research Motivation

The motivation behind this research arises from the persistent challenges faced by individuals with mobility impairments, particularly wheelchair users, in controlling household devices. While offering a degree of autonomy, existing solutions like voice-activated systems and smart glasses have notable limitations, such as reduced effectiveness in noisy environments, physical discomfort, and high costs. Additionally, hardware-dependent systems, such as those utilizing specialized sensors, are often expensive and cumbersome, further restricting accessibility.

We need more natural, intuitive, and affordable solutions that enable individuals with mobility impairments to interact seamlessly with devices. Gesture recognition, especially through pointing direction, offers a promising alternative. However, research on the visual interpretation of 3D pointing gestures is still limited (Nakamura *et al*., 2023). To bridge this gap, this study seeks to develop a machine-learning model that leverages standard RGB cameras for 3D pointing recognition to provide an effective and cost-effective mechanism for interpreting these gestures.

## 1.3 Problem Statement

*Individuals using wheelchairs often face challenges accessing electrical devices, so approximately 35% of U.S. housing units may need to be modified to meet the accessibility requirements of the devices (U.S. Department of Housing and Urban Development, n.d., 2015).*

Even in places that comply with the Americans with Disabilities Act (ADA), individuals using wheelchairs are often required to stretch to reach electrical switches. Furthermore, assistive devices like canes, intelligent eyewear, or mechanical movement aids can be expensive and difficult to maintain.

## 1.4 Thesis Statement

*A two-stage classification system is needed to identify the electrical devices a wheelchair user points at, enabling touchless device control and enhancing accessibility.*

The main output of this research is a device classification system developed in Python to improve accessibility for wheelchair users and support integrators and developers in further improving accessibility. Our research introduces a novel classification system capable of identifying electrical devices pointed to by wheelchair users, utilizing DeePoint (Nakamura et al., 2023) and object detection as key machine-learning technologies. The methodology involves machine learning, transformers, deep learning, computer vision, and object detection. Input data will comprise videos of a seated person pointing at electrical devices, while the output will classify the device and generate a probability score.

## 1.5 Research Objectives

This research primarily aims to create a machine-learning model that classifies devices pointed at by wheelchair users and investigates the accuracy and performance of the classification system across different network architectures and scenarios. The detailed objectives are as follows:

1. Examine the viability of a two-stage machine learning classification system that combines pointing direction prediction with object detection features. Evaluate the system's performance using test data to assess its effectiveness and precision.

2. Examine how the addition of a new input feature (particularly gaze direction) affects the proposed classification system's accuracy and efficiency when compared to the baseline DeePoint model.

3. Analyze the impact of using different object detection models as components within the proposed pointing device classification system, comparing their accuracy and efficiency. Specifically, the study will use YOLO and Fast R-CNN models for comparison.

## 1.6 Research Questions and Hypotheses

This study aims to clarify and explain the following three research questions:

**RQ1:** Can a three-stage classification system be developed to identify the electrical devices a wheelchair user points at, enabling touchless device control and improving accessibility?

**RQ2:** Does tracking gaze direction enhance the accuracy of pointing direction predictions in the third stage of the three-stage classification system?

**RQ3:** Which ML model works best for the third stage of the three-stage classification system for identification of electrical devices, as pointed by wheelchair user?

**H1:** The proposed three-stage classification system can reach 70% accuracy in identifying the electrical devices a wheelchair user points at.

**H1:** Tracking gaze direction can improve the accuracy of pointing direction prediction by approximately 5% in the third stages of a three-stage classification system.

**H3:** In the third stage, the proposed classification system incorporating the Transformer is expected to outperform the model using the MLP.

## 1.7 Scope of Research

The scope of this research is to determine the feasibility of a machine learning system that interprets gesture-based interactions and the capability to classify in-house devices that the person is pointing at. It focuses especially on device accessibility for individuals who use wheelchairs. Some performance measurements include true positive and negative rates, accuracy, precision, recall, confusion matrix, and F1-Score. Training and testing time will also be reviewed as an evaluation factor. The control of the actual electrical devices is not within the scope of this project.

## 1.8 Research Limitations

The following factors limit this research:

While incorporating a confirmation step for pointing gestures would greatly improve reliability and user experience in practical applications, it falls outside the scope of this study. Instead, the focus is on developing a device classification model.

To streamline the model development and ensure focused experimentation, the device categories have been intentionally limited to three common household items: a TV, a Fan, an air conditioner, and a Lamp. This selection allows for a more controlled evaluation of the model's performance without overcomplicating the training process with an extensive array of devices. However, future research could expand this range to incorporate a broader selection of household items, providing a more comprehensive solution to assist individuals with mobility impairments.

Additionally, this study concentrates on users who use their right hand for pointing, and the experiment is restricted to a single type of device in one test video.

Lastly, the testing environment and data collection are confined to indoor settings, with test data exclusively drawn from videos of seated users. While the model aims to eventually support users with mobility challenges (including those in wheelchairs), the current study does not include data from actual wheelchair users but only seated users. Expanding the study to include real-world data from wheelchair users and varying environments would be a logical next step in future work, helping to further validate and enhance the model's applicability.

## 1.9 Organization of Praxis

This Praxis is structured into five chapters:

Chapter 1 introduces the background and research motivation, followed by the research objectives, questions, and hypotheses. It concludes with the research's scope and limitations.

Chapter 2 offers a review of pertinent literature, starting with the DeePoint paper (Nakamura et al., 2023), which underpins this praxis. Additional sources discuss accessibility challenges for wheelchair users. The WorldPoint paper from CMU provides technical details on implementing the ray-casting algorithm for object intersection (Kim *et al*., 2023), while the MultiNet framework illustrates how to effectively merge multiple models into one (Teichmann *et al*., 2018). The end-to-end human-object interaction detection paper suggests a solution for pointing object detection based on the HOI pattern (Zou *et al*., 2018).

Chapter 3 addresses the data and methodologies used to train the models and test the research hypotheses.

Chapter 4 details the results and analyses using the statistical methods outlined in Chapter 3.

Chapter 5 concludes the praxis by discussing the results and insights. It further highlights contributions through a discussion of knowledge and proposes recommendations for future research in the field.

# Chapter 2—Literature Review

## 2.1 Introduction

The growing number of individuals with mobility impairments has highlighted the need for effective assistive technologies that enable users to control devices with minimal physical effort. Among all assistive technologies, gesture-based controls have emerged as a natural and intuitive method for human-device interaction, allowing users to control devices through simple motions (Islam, M.M., 2020). This chapter provides a comprehensive literature review of topics related to assistive technologies, finger-pointing gesture recognition, transformers, and neural network algorithms, which form the core of intelligent assistive technology for mobility-impaired users. The purpose of this review is to summarize the published research on these topics and to analyze the existing body of technical knowledge.

The chapter starts by examining the challenges faced by mobility-impaired individuals and the pressing need for assistive technologies designed to improve their independence and quality of life. It then presents a detailed analysis of existing solutions for these users, drawing from a wide range of literature. The chapter also explores recent advancements in key areas such as gesture recognition, object detection, and human-object interaction. These critical components enable seamless interaction between users and their environments.

Additionally, this chapter reviews the latest research on AI and neural networks, specifically highlighting the transformer and convolutional neural network (CNN) architectures utilized in this field. The discussion centers on their application in gesture-based control systems and object detection tasks. The chapter concludes by summarizing key literature findings and exploring the implications of this research for advancing future assistive technologies, especially regarding improved accessibility for users with mobility impairments.

## 2.2 Assistive Living and Technologies Review

America's aging population is undergoing unprecedented growth, and a significant portion faces mobility challenges, with many requiring wheelchairs to move around and access daily utilities. As individuals age, they are increasingly likely to experience mobility disabilities, which pose serious challenges to independent living. Many residences in the U.S. are not equipped for such needs, often requiring costly modifications to meet accessibility standards. The financial burden of these adjustments is significant. In addition to housing modification, according to the U.S. Department of Health and Human Services (HHS), nearly 70 percent of people who reach age 65 will require some form of long-term care in their lifetime. This additional care and housing modifications can be overwhelming, particularly for those already facing financial constraints.

Compounding this issue is the shrinking number of available caregivers at a time when the demand for long-term care is surging (Nora *et al*., 2020). With fewer caregivers to provide assistance, new solutions are urgently needed to bridge the gap. Technology has the potential to play a transformative role in addressing these challenges. Innovations in assistive technology can significantly improve the quality of life for older adults by providing them with the tools they need to achieve independent living.

For instance, Chen, W. L. et al. introduced a novel home appliance control system tailored for individuals with disabilities, enabling them to perform daily tasks independently. Such systems represent a step toward greater autonomy for people with mobility issues, allowing them to control household devices with minimal physical effort. Another example is the work of Bourbakis, N.G., 2022 who proposed an intelligent system that integrates robots, sensors, and other assistive technologies to aid with mobility. While this system provides a comprehensive solution for those with severe mobility impairments, it is prohibitively expensive for widespread adoption. The combination of advanced robotics, artificial intelligence, and sensor technologies may only be feasible for those with considerable financial resources or specialized needs.

As with many technological solutions, these advances come with their own set of challenges. The primary hurdles involve the cost of these systems (as mentioned) and the obtrusiveness of the devices. High upfront costs can put these technologies out of reach for many older adults – particularly those on fixed incomes. Furthermore, the physical presence of devices in the home can be obtrusive, potentially disrupting the comfort and aesthetics of the living environment. For a practical application in solving accessibility issues for older adults, the challenge lies in developing affordable and unobtrusive technologies while still retaining their effectiveness. Solutions must focus on functionality and user experience, ensuring that devices blend seamlessly into the home environment without being overwhelming or difficult to use. Affordability is also key, especially as the population ages and the number of individuals needing assistance continues to rise.

In response to these challenges, intuitive, low-cost solutions such as gesture recognition systems or simple interfaces to allow individuals to control devices through natural interactions could be designed with AI technology and without the need for complex hardware or invasive modifications (Islam, M.M., 2020). These systems could provide a cost-effective alternative, allowing older adults to easily manage their daily tasks while avoiding the high costs associated with robotics and sensor-heavy systems. By focusing on accessible, affordable, and unobtrusive technological solutions, we can help bridge the gap between the increasing need for care and the dwindling number of caregivers. These technologies have the potential to empower older adults, enabling them to live more independently and with greater dignity as they age. (Courtney, K. L. et al, 2007; Moon NW, et al, 2019)

## 2.3 Pointing Gesture Recognition and Object Interaction

Gesture control has been widely adopted in the augmented and virtual reality (AR/VR) industry and can serve as an ideal daily solution for individuals with mobility impairments. Egocentric vision (also known as first-person vision) usually refers to capturing and processing images and videos from cameras worn on a person’s head. With the development of smart wearable cameras and augmented reality headsets such as Meta Oculus, Microsoft HoloLens, and Google Glass, egocentric vision and its potential applications have drawn much attention. The 2016 CVPR paper, “A Pointing Gesture-Based Egocentric Interaction System: Dataset, Approach, and Application” (Huang *et al*., 2016) studies AR-based pointing technology, especially hand gesture-based interaction. This paper presents a solution for pointing-gesture interaction in egocentric vision and its applications. Firstly, a dataset is established, focusing on pointing gestures for egocentric vision. Second, they propose a two-stage Faster R-CNN-based hand detection and dual-target fingertip detection framework. Later, Cao *et al*. proposed an egocentric gesture recognition using a recurrent CNN with spatiotemporal transformer modules for wearable AR device movement problems. Alam M. M. *et al*. introduce a unified learning approach to predict both the probabilistic output of the egocentric gesture of fingers and the positional output of all the fingertips using one forward propagation of a CNN. For a special hardware example, G. Park et al. developed a gesture recognition method with a radar-antenna system and a deep learning model.

A wearable solution for gesture recognition may pose significant challenges for mobility-impaired individuals, as it can be expensive and physically intrusive. In contrast, Nakamura et al. (2023) offer a more practical approach. They developed a non-wearable solution, called DeePoint, that focused on automatic visual recognition and direction estimation for pointing gestures. This research presents a groundbreaking method for neural-based pointing recognition and introduces the first large-scale dataset designed specifically for this purpose. The DeePoint (DP) Dataset consists of over 2 million frames collected from 33 individuals exhibiting a variety of pointing styles. Each frame has been meticulously annotated with pointing timings and 3D pointing directions, enabling detailed and accurate gesture analysis. This rich dataset serves as a critical resource for training and evaluating models in visual gesture recognition. Additionally, this paper introduces the DeePoint model, the first neural network architecture specifically designed to understand pointing gestures and estimate their direction. Through extensive experimentation, the model demonstrated both high accuracy and efficiency, showcasing its potential for real-world applications. The combination of the DeePoint model and the DP Dataset offers a robust foundation for future advancements in visual human-intention understanding, particularly for non-wearable gesture-based interaction systems. DeePoint research addresses the limitations of wearable technologies and advances nonintrusive solutions, setting the stage for more accessible and intuitive assistive technologies for users with mobility impairments.

WorldPoint (Kim et al., 2023) introduces an innovative concept that leverages pointing gestures for quick and intuitive mobile interactions. This approach eliminates the need for wearable devices, although users are required to always have their mobile phones with them. Developed by Kim et al. at CMU, this technology harnesses recent advancements in wide-angle, rear-facing smartphone cameras combined with hardware-accelerated machine learning, facilitating real-time, infrastructure-free finger-pointing interactions on modern mobile devices.

Finger-pointing occasionally needs verification to fully understand the user’s intent. Constantin et al. propose an error correction technique utilizing natural language in conjunction with pointing gestures. Their approach employs 2D detection of hands and objects, relying on user utterances to rectify mistakes stemming from the misclassification of pointed objects. Xie et al. introduce the Multimodal Guidance Module (MGM), which integrates various input methods, including language directives, pointing gestures, and clicks to identify target areas and sample points as centers of regions. Our research embodies this concept by using two types of inputs, images and pointing direction vectors, to the transformer for device classification.

Recently, pointing gesture technologies have become popular in the automobile industry. In the paper "You Have a Point There: Object Selection Inside an Automobile Using Gaze, Head Pose, and Finger Pointing" (Aftab *et al*. 2020), finger-pointing technology is explored for automotive user interaction. The automotive industry is rapidly advancing in user interaction technologies, with mid-air gestures and voice commands already enhancing driver-vehicle interaction (see Figure 2). This paper proposes a multimodal fusion method: gaze, head pose, and finger-pointing gestures, using speech solely as a trigger for the fusion process. This paper compares state-of-the-art deep neural network architectures with traditional machine learning; the results indicate that deep learning methods can improve pointing direction accuracy when integrating multiple modalities. This multimodal approach has the potential to enhance user interaction in vehicles, establishing a basis for future applications that rely on sensor fusion for a more intuitive and responsive driving experience.

In contrast to other approaches that rely on specialized and often costly gesture camera hardware, our research leverages a software-based solution using the DeePoint model (Nakamura 2023). This shift away from hardware dependency represents a significant advancement, as it allows us to achieve accurate pointing gesture recognition without the need for expensive, dedicated gesture-tracking equipment. By focusing on software-based innovations, we can lower the overall cost of the system, making it more accessible and practical for widespread use, particularly for individuals with mobility impairments who may benefit from affordable assistive technology.

Our software-driven approach capitalizes on the strength of neural networks and advanced algorithms to replicate the functionality typically associated with high-end hardware solutions. By using DeePoint, we maintain high accuracy in gesture recognition and direction estimation, while drastically cutting the expenses that would otherwise be incurred by utilizing specialized cameras. This not only makes the technology more cost-effective but also enhances its portability and ease of implementation in various real-world environments, including homes, healthcare facilities, and public spaces.

Ultimately, this software-first strategy aligns with our goal of developing accessible assistive technologies, reducing barriers for users, and enabling broader adoption across diverse settings. By eliminating the need for dedicated hardware, we pave the way for more scalable and flexible solutions that can integrate seamlessly into existing systems, benefiting a wider range of users without the burden of high costs.

Roider et al. have also presented the problem of object selection inside a car. They integrate eye gaze with finger-pointing gestures using a simple rule-based fusion approach. They have shown that selecting an object on an in-vehicle display screen achieves increased pointing accuracy over a single modality, finger-pointing (Roider et al., 2018). However, this experiment is limited to only four adjacent objects on a screen.

A finger pointing at a car dashboard

Description automatically generated

Figure 3.Driver makes a pointing gesture to interact with the car (Source: Aftab 2020)

In the robotics industry, gestures are a common way for human-robot interaction as studied by the 2023 ECCV paper "Interactive Multimodal Robot Dialog Using Pointing Gesture Recognition" (Tanada et al., 2024). This work proposes a system for interactive, multimodal, task-oriented robot dialog that leverages pointing gesture recognition. The system integrates state-of-the-art computer vision techniques to recognize objects, hand positions, orientations, and overall human poses, allowing for a comprehensive understanding of pointing gestures and the corresponding target objects. Furthermore, M. Ürkmez and H. I. Bozma propose a two-stage CNN approach to detect 3D hand-pointing direction. However, this method requires a depth camera and focuses on HRI (human-robot interaction) applications. Like their use with robots, pointing gestures are also used in drone applications, as in the paper “Using Pointing Gesture to Define a Target Object” (Medeiros et al. 2020), which develops a method for firefighters to specify a fire’s location and direct drones to fly to it. A similar paper by Medeiros et al. integrates depth info with simultaneous localization and mapping (SLAM) to achieve the same purpose.

Human-object interaction (HOI) detection is a crucial component in advanced human-centric scene understanding, and it has garnered significant research attention in recent years. The primary objective of HOI detection is not only to localize humans and objects within a scene but also to accurately recognize their interactions. This capability is essential for applications such as robotics, autonomous systems, and assistive technologies, where understanding human intentions and actions in relation to objects is key.

Previous research, such as the work by Chen Gao (2018; 2020) has produced promising results by employing a two-stage approach to HOI detection. In these studies, the task is decomposed into two steps: object detection and interaction classification. This method first identifies the objects and humans separately and then classifies the type of interaction occurring between them. Although effective, this two-stage process can introduce inefficiencies and unneeded complexity.

More recent methods such as those proposed by Tiancai Wang, Kim B., and Yue L. have advanced HOI detection by formulating a surrogate interaction detection problem. These approaches aim to optimize HOI detection indirectly, using a more streamlined one-stage approach. This method combines object detection and interaction recognition into a single step, improving efficiency and potentially increasing detection accuracy by allowing for a more holistic analysis of the scene.

Additionally, Zou et al. introduce a groundbreaking end-to-end HOI prediction model that eliminates the need for multiple stages entirely. Their approach enables direct prediction of human-object interactions, simplifying the process and improving the overall detection efficiency. This end-to-end approach holds particular promise for applications in gesture-object interaction, where recognizing gestures in relation to objects in real time is critical. Such advancements are paving the way for more effective and seamless interaction systems, especially in environments requiring a quick and accurate understanding of human behavior, such as assistive technologies or gesture-based control systems.

## 2.4 Vision Transformer Technology

This section provides an overview of vision transformer (ViT) technology and references the key papers related to its development.

The core of transformer technology is the attention mechanism, which helps machine learning models focus on specific parts of the input data rather than processing all information equally. It allows the model to "attend" to the most relevant pieces of data when making predictions. In sequence models like transformers, attention calculates the importance of each word or token in relation to others, enabling the model to capture long-range dependencies and contextual relationships more effectively. In other words, an attention function can be understood as a process that takes a query and a set of key-value pairs as inputs and generates an output. Each of these (query, keys, values, output) is represented as a vector. The output is a weighted sum of the values, where the weight for each value is determined by how closely the query matches the corresponding key, computed through a compatibility function. This allows the model to focus on relevant information in the data (Vaswani et al., 2017), making the attention mechanism highly useful in natural language processing and image recognition tasks.

The attention-based transformer architecture has become the de facto standard for natural language processing tasks, but its applications to computer vision remain limited. In vision, CNNs remain dominant (LeCun et al., 1989; Krizhevsky et al., 2012); focus is either applied in conjunction with convolutional networks or used to replace certain components of convolutional networks while keeping their overall structure in place. Dosovitskiy et al. at Google Brain published a paper titled “An image is worth 16x16 words: Transformers for image recognition at scale” (Dosovitskiy et al., 2020), which expanded the previously NLP-only usage of transformers by introducing the vision transformer (see Figure 3). This paper explores directly applying transformers to images by treating image patches as tokens for classification. In our proposed architecture, we have integrated transformer technology by leveraging a similar vision transformer architecture.

A diagram of a bird ball car

Description automatically generated

Figure 4.Vision Transformer Architecture (Source: Dosovitskiy 2020)

The evolution of architectures for video understanding has closely followed the progress made in transformer-based models for image recognition. One significant contribution to this area is the video vision transformer (ViViT), as presented in the work by Arnab and colleagues in 2021. In this paper, the authors developed pure transformer-based architectures specifically designed for video classification tasks, drawing inspiration from the success of the vision transformer in image processing. The motivation behind utilizing transformer architectures for understanding video stems from the inherent ability to model long-range dependencies and capture contextual relationships over time, which are crucial for analyzing video data.

Transformers – particularly their self-attention mechanisms – excel at understanding data sequences by focusing on different parts of the input simultaneously and learning how elements relate to one another. In video processing, where the input consists of frames over time, these attention-based architectures are particularly well-suited for modeling both the temporal and spatial dimensions of video. The video vision transformer leverages this capability by extending the transformer’s attention mechanism to the spatial features of individual video frames and the temporal dependencies between consecutive frames, enabling the model to capture rich contextual information that spans time and space.

By using a pure transformer approach rather than relying on convolutional neural networks or recurrent architectures, the model can process video data more holistically, considering global relationships within the video stream. This is especially useful for tasks that require understanding complex actions or interactions over time, such as activity recognition, event detection, or video-based object tracking.

Arnab et al.'s work represents a key advancement in video classification, as it demonstrates how transformer-based models originally designed for static images can be adapted to handle the dynamic and sequential nature of video. The introduction of the video vision transformer not only highlights the versatility of transformers in various computer vision tasks but also sets a new standard for how we approach video understanding. By leveraging the transformer’s attention mechanisms, the architecture is better equipped to manage the complexities inherent in video data, such as long-range temporal dependencies and high-dimensional input, offering a more intuitive and powerful framework for video analysis.

To integrate text with images in transformer-based models, CLIP (Contrastive Language-Image Pretraining), developed by Radford and colleagues in 2021 at OpenAI, provides a groundbreaking approach that connects natural language with visual understanding. CLIP is trained on an extensive dataset consisting of text-image pairs and employs contrastive learning to align visual inputs with their corresponding text descriptions. This enables CLIP to perform tasks such as zero-shot image classification, where the model can match images to relevant labels without requiring task-specific fine-tuning. CLIP’s ability to bridge language and visual content represents a major advancement in multimodal AI, offering a flexible and powerful framework for applications ranging from image retrieval to visual context understanding based on textual descriptions.

The success of CLIP has led to various adaptations, such as CLIP2, developed by Zeng et al. This version expands CLIP’s functionality into the 3D domain, learning transferable 3D point cloud representations for real-world applications. By utilizing a novel proxy alignment mechanism, we broaden CLIP's scope beyond just 2D image-text tasks to also encompass 3D object recognition and interaction. These developments underline the adaptability and potential of CLIP-based models to enhance linguistic and visual understanding in a variety of contexts.

## 2.5 Object Detection Technology

This section provides a comprehensive overview of object detection technology and highlights key contributions from foundational papers in the field.

Object detection has been a core challenge in computer vision for decades, with a wide range of applications in areas such as image understanding, robotics, and autonomous systems. The development of effective object detection algorithms has been critical for enabling machines to perceive and interact with their environments. One of the most influential advancements in this area is the YOLO (You Only Look Once) algorithm, first introduced by Joseph Redmon and colleagues in 2016 (Redmon et al., 2016). YOLO revolutionized object detection by departing from traditional region-based approaches, which rely on generating proposals for possible object locations within an image. Instead, YOLO frames object detection as a single-stage regression problem, allowing it to predict both the object class and its bounding box coordinates in one pass through the neural network (see Figure 4).

The YOLO algorithm has undergone significant evolution since its initial release, with researchers introducing more refined and optimized versions over the years. These include YOLOv2, YOLOv3, YOLOv4, and YOLOv5, each improving on the model’s speed, accuracy, and ability to detect smaller objects (Zhao et al., 2019; Laroca et al., 2018). Additionally, lighter versions such as YOLO-LITE have been developed to make the model more suitable for resource-constrained environments, enabling real-time detection on devices with limited computational power (Huang et al., 2018).

By October 2024, the YOLO framework has reached its 11th version, continuing to push the boundaries of real-time object detection. These advancements reflect ongoing efforts in the research community to balance accuracy and efficiency in various applications, from self-driving vehicles to surveillance and robotics. Each version of YOLO has built on the strengths of its predecessors, incorporating new techniques and technologies to maintain its position as one of the most widely used and impactful object detection algorithms in the field.

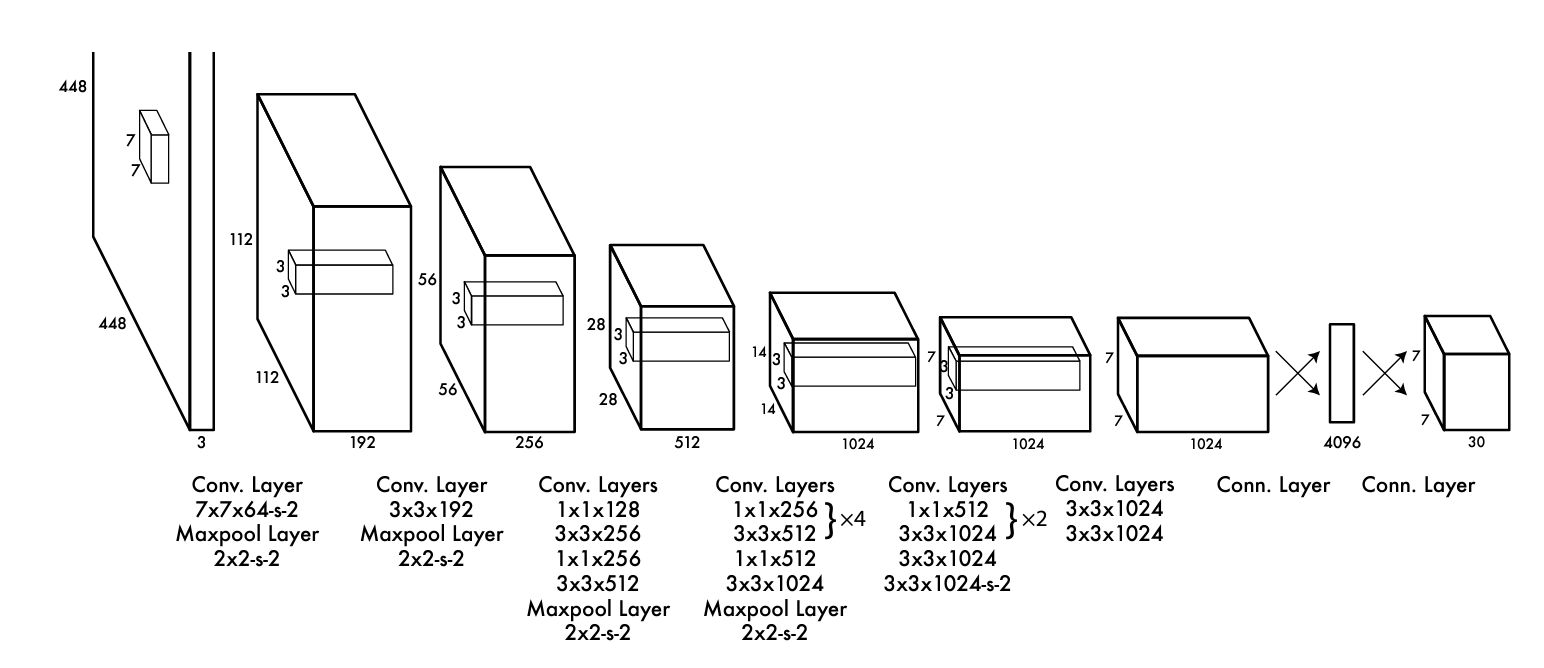


Figure 5. YOLO architecture (Source: Redmon et al., 2016)

Another widely adopted object detection algorithm is Fast R-CNN (Figure 6), developed by Ren and colleagues in 2016. This algorithm builds on the limitations of earlier region-based object detection methods by introducing a more efficient, integrated approach. Fast R-CNN is composed of two key modules that work together to enable faster and more accurate detection. The first module is a deep, fully convolutional network that generates region proposals, which are potential areas in the image that may contain objects. This process of generating candidate regions helps narrow down the areas that require further analysis, significantly reducing the computational load.

The second module is the Fast R-CNN detector, which processes these proposed regions to classify the objects and refine the bounding boxes. Unlike earlier systems that required multiple stages to accomplish these tasks, Fast R-CNN unifies both modules into a single, end-to-end trainable neural network. This seamless integration allows for joint optimization of both region proposals and object classification, leading to improved detection accuracy while maintaining a high inference speed.

A key feature of Fast R-CNN is its ability to handle the entire detection pipeline in a single forward pass, making it faster than previous models like R-CNN and SPPnet. In addition, the algorithm leverages modern techniques found in neural networks with attention mechanisms (Chorowski et al., 2015). These mechanisms enable the model to focus on relevant parts of the image, enhancing its ability to detect objects that may otherwise be overlooked. Attention mechanisms are particularly useful for dealing with complex scenes where multiple objects are present or when the objects are small or partially obscured.

Fast R-CNN’s combination of region proposal generation and efficient object detection has made it a foundational technique in the field of computer vision, influencing subsequent developments in object detection models. By streamlining the process into a single unified network, Fast R-CNN set a new standard for performance and efficiency, contributing to the advancement of real-time object detection across various applications, such as autonomous driving, video analysis, and robotic perception. The innovation of Fast R-CNN continues to be referenced in modern object detection research, illustrating its lasting impact on the field.

A diagram of a person

Description automatically generated

Figure 6: Fast R-CNN architecture (Source: Ren et al., 2016)

## 2.6 Summary and Conclusion

This literature review offers an in-depth analysis of scholarly research focused on key areas such as assistive technology, gesture recognition, finger-pointing technology in industrial applications, and transformer-based models for device classification in pointing tasks. Through this examination, several trends and insights have emerged regarding the role of finger-pointing as a natural and effective means of interacting with objects in both virtual and physical environments.

Finger pointing has proven to be an intuitive method for users to specify or select objects across a wide range of applications. In augmented reality, robotics, drones, and automotive interfaces, among other situations, finger-pointing simplifies interaction by allowing users to naturally direct attention or control devices through gestures. Several industries have integrated this technology to enhance user experience and operational efficiency. For instance, automotive companies are increasingly exploring finger-pointing for in-car interfaces, enabling drivers to control dashboard functions without manual input. Similarly, AR systems benefit from finger-pointing by allowing users to interact with virtual objects in immersive environments with greater ease.

Importantly, the literature also highlights the potential for these same technologies to be adapted for the assistive technology sector, particularly in supporting individuals with mobility impairments. For people who face challenges in performing everyday tasks, finger-pointing combined with gesture recognition and device classification systems presents an opportunity to regain independence. By leveraging transformers and other advanced models, these systems can enable users to interact with home devices, computers, or other assistive systems through simple gestures, bypassing the need for more physically demanding interfaces.

The review underscores the versatility of finger-pointing technology, showing its relevance not only in commercial and industrial applications but also in its capacity to revolutionize assistive technology solutions. Through the use of transformer-based models, the same gesture-based systems which enhance AR and robotic technology can be repurposed to provide the intuitive, low-effort control systems which would prove so significant – not only to individuals with mobility impairments, but to any individual responsible for controlling an automotive system, digital application, or smart device. This in turn stands to open the door for more inclusive technologies going forwards, giving people of all kinds the opportunity to live more independently and more naturally within their environment.

# Chapter 3—Methodology

## 3.1 Introduction

This chapter outlines the methodology used in this study, focusing on the development and evaluation of a three-stage classification system designed to identify the devices an individual is pointing at. Several data processing methods and models have been developed or integrated. The chapter begins with an overview of the data collection process, highlighting the use of cloud data storage and the gathering of the training dataset (DeePoint, Nakamura, 2023). It then transitions to data preprocessing techniques, including formatting and filtering the data. The next phase concentrates on the three-stage architecture (Figure 8), which involves Stage-1, estimating the pointing direction, followed by Stage-2, dedicated to device localization. The chapter then explores the development of a transformer that serves as the Stage-3 model, capable of evaluating the vector alignment between the outputs from Stages 1 and 2. The results from Stage 3 can be used as the basis for decisions regarding the device class. An effort is being made to assess the inclusion of the gaze direction feature in Stage-3 to determine whether it could potentially provide enriched contextual information to improve the model’s accuracy. In addition, the performance of the Stage 3 model is evaluated against various machine learning architectures. Finally, the overall system is tested using generated videos to assess its performance with unseen data.

A diagram of a process

AI-generated content may be incorrect.

Figure 7: End-to-end Process

## 

A screen shot of a diagram

AI-generated content may be incorrect.

Figure 8: Three Stages Architecture

## 3.2 Collect data

### 3.2.1 Configure AWS Infrastructure for Data Storage and Processing

The AWS infrastructure was configured for data storage by creating Amazon S3 buckets to hold datasets, enabling versioning, and establishing lifecycle policies to optimize costs by transitioning older data to more economical storage classes, such as S3 Glacier. All services were set up to ensure secure access using IAM roles, bucket policies, and server-side encryption. Additionally, serviceswere monitored through AWS CloudTrail and S3 Access Logs. To support file-based datasets that require shared access or high throughput, Amazon Elastic File System (EFS) was deployed with the appropriate performance and throughput modes, and access was controlled through security groups and NFS access points.

Additionally, disaster recovery was implemented by scheduling automatic backups with AWS Backup and establishing cross-region replication for S3. To facilitate data processing, S3, and EFS services were integrated with computing services such as SageMaker, Lambda, and EC2 for seamless data access during processing or training. Finally, storage usage and costs were monitored using tools like AWS Cost Explorer and S3 Storage Lens. Tagging and storage tiers like S3 IntelligentTiering were employed to optimize expenses. This setup ensures scalable, secure, and cost-effective data management and storage.

### 3.2.2 Datasets for Model Training

The DeePoint dataset was specifically designed to train models aimed at understanding human-device interactions, including pointing gestures, gaze alignment, and contextual awareness in 3D spaces. This data comprised images or video frames capturing pointing gestures, 3D spatial coordinates of critical points such as fingertip positions, and reference markers on target objects. We also incorporated additional modalities, like depth maps or gaze tracking data, to provide richer contextual information.

The training dataset was designed to encompass a variety of scenarios, including different lighting conditions, angles, and user behaviors, ensuring robustness during model training. Ground truth annotations, such as labeled target objects or directional vectors, are crucial for supervised learning tasks.

The DeePoint dataset has been extracted and organized as illustrated in the table below (Table 1), facilitating the training and assessment of the DeePoint model, with the sample frame depicted in Figure 9. In this process, we took the DeePoint dataset and underwent a data processing step to create a dataset for training the Stage-3 model. ( refer to 3.6.2)

|  |  |
| --- | --- |
| Data Column | Data Column Information |
| date | Datetime |
| venue | Date and place, e.g. 2023-01-2-17-office |
| session | Full recording of a scene. e.g. Take1, take2 |
| camera\_id | Camera identifier |
| keypoints | Person’s body, hands… keypoints |
| frame | Image frame, e.g. 00000001.jpg |
| marker id | Reference ArUco marker |

Table DeePoint Data Sample

A person standing in a room

AI-generated content may be incorrect.

Figure 9: DeePoint Training Image Example

#### 3.2.3 Data Visualization

A 3D DeePoint data visualization tool using a Python notebook was developed to improve understanding of the DP dataset's structure. The blue dots represent ArUco markers, while the red dots indicate the cameras. It also displays a person’s body poses in the rendering (Figure 10).

A computer generated image of a person

Description automatically generated

Figure 10: DeePoint 3D data visualization

## 3.3 Preprocess Data

These were the steps to prepare the raw data for later-stage training purposes:

* Data cleaning and filtering: Eliminate frames without a valid pointing gesture from the DeePoint dataset. A file lists the frame numbers of valid pointing gesture images, which can be used to carry out the necessary cleaning and filtering.
* Data loading logic: Create a Python class and implement the logic to enable efficient data loading for training purposes.
* Data Split: Separate the dataset into training, evaluation, and test sets, allocated as 80%, 10%, and 10%.
* Pointing Vector Processing: For DeePoint training, the ground truth pointing direction vector is determined from the person’s fingertip keypoint to the marker being pointed at, as shown in Figure 11.

A silhouette of a person pointing at a point

AI-generated content may be incorrect.

Figure 11: Pointing Direction Vector

## 3.4 Perform pointing direction estimation (Stage-1)

Stage-1 is powered by the DeePoint model (Nakamura 2023), introduced in the 2023 DeePoint paper. This model utilizes a transformer-based network to determine the pointing direction by tracking the user’s body, hand, and finger poses from the video input. It estimates a high-fidelity 3D pointing vector along with a confidence score. DeePoint is integrated into Stage-1 without any modifications or fine-tuning.

### 3.4.1 Stage-1 Architecture

A diagram of a computer program

AI-generated content may be incorrect.

Figure 12. Stage-1 Architecture

### 3.4.2 Deploy Stage-1

The DeePoint model was established on our AWS account as is, without any adjustments or fine-tuning. This model serves as the initial stage of the pipeline, providing the pointing direction estimation function for later use. To verify the installation, we created videos demonstrating that the deployed model can accurately generate the pointing direction vector.

## 3.5 Perform device detection & localization (Stage-2)

The second stage aims to process the image input and produce the device location unit vector along with the device class, as shown in Figure 13.

A diagram of a device model

AI-generated content may be incorrect.

Figure 13. Stage-2 Architecture

#### 3.5.1 Object Detection Model

Detecting objects in an image involves identifying and locating specific items of interest within a broader visual context. In this stage, the YOLO model is employed to detect devices in the scene. We limit the types of objects to detect to a TV, lamp, and fan in this practice. The outcome of the object detection includes the object class, which will be referenced in the final decision.

#### 3.5.2. Depth Estimation Model

The primary reason for adopting a depth estimation model is to convert 2D coordinates into 3D for subsequent vector computation. A depth value for the object is estimated either from depth sensors or through monocular depth estimation techniques. This depth value provides the missing third dimension essential for positioning the object in 3D space alongside the 2D coordinates. The normalized 2D coordinates are combined with the depth information and back-projected into 3D space using the camera's intrinsic and extrinsic parameters. This process yields the 3D corners or edges of the bounding box relative to the camera's coordinate frame.

#### 3.5.3 3D Localization Module

The process of converting 2D to 3D coordinates involves transitioning from a flat, two-dimensional representation of an object in an image to a more comprehensive, three-dimensional representation that reflects the object’s spatial geometry in a physical environment. A 2D bounding box is typically defined by its top-left and bottom-right coordinates (or similar pairs) in pixel space, encapsulating the object within a rectangular region on the image plane. However, this representation lacks depth information, which is critical for understanding the object's actual size, orientation, and position in the real world.

To perform the conversion, additional information, including camera parameters (e.g., focal length, intrinsic/extrinsic matrices) and depth data (from LiDAR, stereo cameras, or depth sensors), is required. The typical steps include:

**Projection Mapping**: Utilizing the camera's intrinsic parameters, the 2D coordinates were translated into normalized image coordinates. This process converts the pixel-based representation into a format compatible with the 3D coordinate system.

**Dimension and Orientation Estimation**: The 3D bounding box was further refined by estimating its size (height, width, and depth) and orientation (rotation angles) to better align it with the object's actual shape in the physical world. This is often accomplished using prior knowledge of object categories, machine learning models, or optimization techniques.

## 3.6 Train model to classify pointing-device alignment (Stage-3)

### 3.6.1. Stage-3 Model Architecture

Stage-3 is designed from the ground up to classify pointing-device alignment using integrated vectors of pointing direction and device location. This model is based on a Transformer architecture, and we also evaluate various ML algorithms for comparative analysis (Figure 14).

A diagram of a transformer encoder

AI-generated content may be incorrect.

Figure 14. Stage-3 Architecture

### 3.6.2 Training dataset

The Stage 3 training dataset is derived from the DeePoint dataset (Table 1), from which we selected a subset containing 90,000 rows and processed the data to compute various direction vectors. This dataset is split into 80% for training, 10% for validation, and 10% for testing. The model's inputs include a pointing direction vector and a device location vector. These vectors are computed from the DeePoint dataset during training: the pointing vector is determined from a person’s fingertip keypoint to the marker, while the device location vector is calculated from the person’s center keypoint to the marker (Table 2).

|  |  |
| --- | --- |
| Data Column | Data Column Information |
| date | Datetime |
| site | Place, e.g. office, living room |
| frame | Image frame, e.g. 00000001.jpg |
| pointing\_direction | Pointing gesture direction vector |
| device\_direction | Device location direction vector |
| gaze\_direction | Gaze direction vector (for comparison) |
| is\_aligned | Are vectors aligned or not |

Table Stage-3 Training Data

### 3.6.3 Model Training

The training data for the third stage was primarily derived from DeePoint, but only a specific subset of this data is needed. This dataset includes three key types of information: object images, which provide visual representations of the targets; pointing direction vectors (calculated from the DeePoint dataset), which indicate the direction of focus or interest; and location coordinates, which specify the precise spatial or geographical positions (from first stage maker position data). Together, these elements form the core inputs necessary for effective training in the second stage.

The third **stage** transformer model processes spatial information as three-dimensional pointing vectors and normalized positional coordinates. These features are projected into a shared embedding space through fully connected (dense) layers. Each dense layer generates fixed-size feature vectors, facilitating their integration with the image embeddings. The projected embeddings are normalized to unit vectors to ensure consistency and stability during training.

The model introduces an innovative approach to feature fusion by employing a weighted combination of image, pointing vector, and position features. Three weights (α1, α2​) - hypermeters determine the relative importance of each feature type, allowing the model to adapt dynamically to the dataset's requirements. The combined feature vector undergoes additional dropout for regularization and is reshaped to match the input format the subsequent transformer encoder expects. The equations are shown as follows:

#### 3.6.3.1 Transformer Encoder and Multi-head Self-attention

The transformer encoder, a crucial component of the model, learns the contextual relationships among the combined features. The encoder captures complex dependencies across feature dimensions through multi-head self-attention, allowing the model to grasp nuanced interactions between visual and spatial information.

#### 3.6.3.2 Loss Function

Two loss functions are combined during training. The first is the Binary Cross-Entropy Loss (BCELoss), which serves as a loss function for binary classification.

The equation is presented as:

The second loss function is the cosine similarity between the pointing vector and the device location vector. The equation is presented as follows:

#### 3.6.3.3 Optimizer

We used the Adam (Adaptive Moment Estimation) optimizer, which adjusts the learning rate individually for each parameter by using estimations of the first moment (mean) and the second moment (uncentered variance).

#### 3.6.3.4 Learning Rate Scheduler

The **ReduceLROnPlateau** scheduler can be applied during training. This type of learning rate scheduler is commonly used in training machine learning models to enhance the training process. Its primary purpose is to adjust the learning rate dynamically based on the model's performance throughout training. Specifically, it monitors a performance metric, such as validation loss or accuracy, and reduces the learning rate when the metric stops improving for a specified number of epochs, known as the **patience** parameter. This approach helps the model converge more effectively by utilizing a higher learning rate when improvements are frequent and switching to a lower learning rate when the training process reaches a plateau.

#### 3.6.3.5 Outputs

The encoder’s output was then passed to a classification head and implemented as a fully connected layer. This head produced logits corresponding to the predefined classes. The logits equation is shown as follows:

A Sigmoid layer converts these logits into probabilities, producing interpretable outputs for classification tasks. The equation for Sigmoid probabilities is as follows:

#### 3.6.3.6 Summary

This architecture exemplifies a synergistic integration of spatial feature embedding and contextual modeling (transformer encoder). Its modular design, along with the incorporation of normalization, dropout, and weighted feature fusion, ensures robustness, scalability, and flexibility for a wide array of applications involving 3D pointing and object recognition.

## 3.7 Evaluate how gaze direction impacts accuracy

## 3.7.1 Incorporate the gaze direction feature in Stage-3

In Stage-3, we evaluate the alignment between the pointing vector and the location vector to assess whether a person is aiming at a device. The question arises: will incorporating gaze direction as a third input improve the accuracy? To answer this question, we add gaze direction as the third input to Stage-3 model and evaluate the performance.

### 3.7.2 Modify the model inputs to include the gaze direction feature

To further improve the performance of the existing Stage-3 model, gaze direction is incorporated into the input before being fed into the transformer encoder, resulting in the following combined embeddings:

### 3.7.3 Re-Train Stage-3

After the gaze direction loss function was added to the DeePoint model, we retrained it using the DeePoint training dataset described in earlier sections to evaluate its performance.

## 3.8 Assess the impact of model choice on Stage-3 outcomes

### 3.8.1 Setup system

Set up the AWS cloud to measure the improvements of the fine-tuned DeePoint model, which integrates gaze direction loss with the DeePoint training dataset described in earlier sections.

## 3.8.2 Run Experiment with MLP Model Architecture

Conduct an experiment using the MLP-based Stage-3 model enhanced without gaze functionality. The experiment should specifically focus on evaluating the effectiveness of incorporating MLP model. Use the DeePoint testing dataset to assess the model's performance under this configuration and compare with Stage-3 baseline performance.

## 3.8.3 Compare Performance with the Baseline Model

Key performance metrics, such as accuracy and loss, were systematically analyzed to ascertain whether the fine-tuned model outperformed, matched, or underperformed the Stage-3 baseline under similar test datasets and conditions.

## 3.9 Build a single pipeline connecting all stages

### 3.9.1. End-to-End Architecture

The end-to-end method used in this practice was designed with a three-stage architecture (as shown in Fig. 10). The first stage detects the pointing gesture and predicts the pointing direction mainly based on the DeePoint paper. The second stage employs a transformer for device classification; the input for the transformer encoder is the concatenation of three embeddings: 1) the pointing direction, 2) the object image embedding, and 3) the object location direction embedding. The output of the classification transformer is the predicted class of the device that the person is pointing at.

### 3.9.2 Setup system

Set up the AWS cloud to evaluate the performance of the three-stage classification model, which is integrated with the DeePoint training dataset discussed in previous sections.

A collage of images of people in a room

Description automatically generated

Figure 15: Sample images from test videos

## 3.10 Evaluate the Performance of the End-to-End pipeline

The praxis evaluated Accuracy, Precision, Sensitivity, and F1-Score as defined below (Berman et al., 2019):

Accuracy: Accuracy is the ratio of the correct predictions of True Positive

(TP) and the True Negatives (TN) of attacks against the total number of tests

cases:

*Accuracy = TP + TN / (TP + TN + FN + FP)*

Precision: Precision measures the accuracy of positive predictions (FP =

False Positive):

*Precision = TP / (TP + FP)*

Recall/Sensitivity: Sensitivity measures the ability for the algorithm to

determine the TP correctly:

*Recall/Sensitivity = TP/ (TP + FN)*

F1-Score: F1-Score measures the ML model’s accuracy. It combines the

precision and recall scores of a model:

*F1-score = 2\* Precision \* Recall / (Precision + Recall)*

## 3.11 Summary

This chapter explores the methodologies and techniques used for gathering, filtering, and preprocessing raw data, laying the foundation for subsequent training and evaluation. It details the steps involved in transforming raw inputs into a clean and structured format suitable for model training. Additionally, the chapter presents the analytical methods employed for hypothesis testing, providing insights into how data-driven conclusions are derived. A significant focus is given to developing a three-stage classification system specifically designed to assist mobile-impaired individuals in identifying the devices they intend to interact with. This system highlights the integration of advanced preprocessing techniques and classification algorithms to enhance accessibility and user experience for individuals with mobility challenges.

# Chapter 4—Results

## 4.1 Introduction

## 4.2 Another Section

# Chapter 5—Discussion and Conclusions

## 5.1 Discussion

## 5.2 Conclusions

## 5.3 Contributions to Body of Knowledge

## 5.4 Recommendations for Future Research

# References

U.S. Department of Housing and Urban Development. (n.d.). Accessibility of America’s housing stock: Analysis of the 2011 American Housing Survey. Office of Policy Development and Research.

Bureau of Labor Statistics, U.S. Department of Labor. (n.d.). *Occupational Outlook Handbook*. Retrieved from <https://www.bls.gov/ooh/>

U.S. Department of Health and Human Services, Office of the Assistant Secretary for Planning and Evaluation. (2022). *What is the lifetime risk of needing and receiving long-term services and supports?* <https://aspe.hhs.gov/reports/what-lifetime-risk-needing-receiving-long-term-services-supports-0>

Nora Super, Three Trends Shaping the Politics of Aging in America, *Public Policy & Aging Report*, Volume 30, Issue 2, 2020, Pages 39-45, <https://doi.org/10.1093/ppar/praa006>

Bourbakis, N.G. (2022). Challenges in Assistive Living Based on Tech Synergies: The Cooperation of a Wheelchair and A Wearable Device. In: Tsihrintzis, G.A., Virvou, M., Esposito, A., Jain, L.C. (eds) Advances in Assistive Technologies. Learning and Analytics in Intelligent Systems, vol 28. Springer, Cham. <https://doi.org/10.1007/978-3-030-87132-1_11>

Chen, W. L., Liou, A. H. A., Chen, S. C., Chung, C. M., Chen, Y. L., & Shih, Y. Y. (2007). A novel home appliance control system for people with disabilities. *Disability and Rehabilitation: Assistive Technology*, *2*(4), 201–206. <https://doi.org/10.1080/17483100701456012>

Islam, M.M., Islam, M.R. & Islam, M.S. An Efficient Human Computer Interaction through Hand Gesture Using Deep Convolutional Neural Network. *SN COMPUT. SCI.* **1**, 211 (2020). https://doi.org/10.1007/s42979-020-00223-x

Courtney, K. L., Demiris, G., & Hensel, B. K. (2007). Obtrusiveness of information-based assistive technologies as perceived by older adults in residential care facilities: A secondary analysis. *Medical Informatics and the Internet in Medicine*, *32*(3), 241–249. <https://doi.org/10.1080/14639230701447735>

Moon NW, Baker PM, Goughnour K. Designing wearable technologies for users with disabilities: Accessibility, usability, and connectivity factors. Journal of Rehabilitation and Assistive Technologies Engineering. 2019;6. doi:10.1177/2055668319862137

Jirak, D., Biertimpel, D., Kerzel, M. *et al.* Solving visual object ambiguities when pointing: an unsupervised learning approach. *Neural Comput & Applic* **33**, 2297–2319 (2021). https://doi.org/10.1007/s00521-020-05109-w

Nakamura, S., Kawanishi, Y., Nobuhara, S., & Nishino, K. (2023). DeePoint: Visual Pointing Recognition and Direction Estimation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision* (pp. 20577-20587).

Huang, Y., Liu, X., Zhang, X., & Jin, L. (2016). A pointing gesture based egocentric interaction system: Dataset, approach and application. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops* (pp. 16-23).

Cao, C., Zhang, Y., Wu, Y., Lu, H., & Cheng, J. (2017). Egocentric gesture recognition using recurrent 3d convolutional neural networks with spatiotemporal transformer modules. In *Proceedings of the IEEE international conference on computer vision* (pp. 3763-3771).

Alam, M. M., Islam, M. T., & Rahman, S. M. (2022). Unified learning approach for egocentric hand gesture recognition and fingertip detection. *Pattern recognition*, *121*, 108200.

Dosovitskiy, A. (2020). An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*.

Yue Liao, Si Liu, Fei Wang, Yanjie Chen, Chen Qian, and Jiashi Feng. Ppdm: Parallel point detection and matching for real-time human-object interaction detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 482–490, 2020.

Kim Bumsoo, Choi Taeho, Kang Jaewoo, and J. Kim Hyunwoo. Uniondet: Union-level detector towards real-time human-object interaction detection. In European Conference on Computer Vision. Springer, 2020.

Tiancai Wang, Tong Yang, Martin Danelljan, Fahad Shahbaz Khan, Xiangyu Zhang, and Jian Sun. Learning human-object interaction detection using interaction points. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 4116–4125, 2020.

Gao, C., Xu, J., Zou, Y., & Huang, J. B. (2020). Drg: Dual relation graph for human-object interaction detection. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XII 16* (pp. 696-712). Springer International Publishing.Chen Gao, Yuliang Zou, and Jia-Bin Huang. ican: Instancecentric attention network for human-object interaction detection. arXiv preprint arXiv:1808.10437, 2018.

M. Ürkmez and H. I. Bozma, "Detecting 3D Hand Pointing Direction from RGB-D Data in Wide-Ranging HRI Scenarios," *2022 17th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, Sapporo, Japan, 2022, pp. 441-450, doi: 10.1109/HRI53351.2022.9889385.

Redmon, J. (2016). You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. Girshick, R. (2015). Fast r-cnn. In *Proceedings of the IEEE International Conference on Computer Vision* (pp. 1440-1448).

Ren, S., He, K., Girshick, R., & Sun, J. (2016). Faster R-CNN: Towards real-time object detection with region proposal networks. *IEEE transactions on pattern analysis and machine intelligence*, *39*(6), 1137-1149.

Chorowski, J. K., Bahdanau, D., Serdyuk, D., Cho, K., & Bengio, Y. (2015). Attention-based models for speech recognition. *Advances in neural information processing systems*, *28*.

Constantin, S., Eyiokur, F. I., Yaman, D., Bärmann, L., & Waibel, A. (2023). Multimodal Error Correction with Natural Language and Pointing Gestures. In *Proceedings of the IEEE/CVF International Conference on Computer Vision* (pp. 1976-1986).

Zou, C., Wang, B., Hu, Y., Liu, J., Wu, Q., Zhao, Y., ... & Sun, J. (2021). End-to-end human object interaction detection with hoi transformer. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 11825-11834).

Kim, D., Mollyn, V., & Harrison, C. (**2023**). WorldPoint: Finger Pointing as a Rapid and Natural Trigger for In-the-Wild Mobile Interactions. *Proceedings of the ACM on Human-Computer Interaction*, *7*(ISS), 357-375.

Dosovitskiy, A. (2020). An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*.

Vaswani, A. (2017). Attention is all you need. *Advances in Neural Information Processing Systems*.

Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., ... & Sutskever, I. (2021, July). Learning transferable visual models from natural language supervision. In *International conference on machine learning* (pp. 8748-8763). PMLR.

Arnab, A., Dehghani, M., Heigold, G., Sun, C., Lučić, M., & Schmid, C. (2021). Vivit: A video vision transformer. In *Proceedings of the IEEE/CVF international conference on computer vision* (pp. 6836-6846).

LeCun, Y., Boser, B., Denker, J. S., Henderson, D., Howard, R. E., Hubbard, W., & Jackel, L. D. (1989). Backpropagation applied to handwritten zip code recognition. *Neural Computation*, *1*(4), 541-551

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, *25*.

Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., ... & Sutskever, I. (2021, July). Learning transferable visual models from natural language supervision. In *International conference on machine learning* (pp. 8748-8763). PMLR.

G. Park, V. K. Chandrasegar and J. Koh, "Hand Gesture Recognition using Deep learning Method," *2021 IEEE International Symposium on Antennas and Propagation and USNC-URSI Radio Science Meeting (APS/URSI)*, Singapore, Singapore, 2021, pp. 1347-1348, doi: 10.1109/APS/URSI47566.2021.9703901.

Zeng, Y., Jiang, C., Mao, J., Han, J., Ye, C., Huang, Q., ... & Xu, H. (2023). CLIP2: Contrastive language-image-point pretraining from real-world point cloud data. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 15244-15253). Zhou, Kanglei & Chen, Chen & Ma, Yue & Leng, Zhiying & Shum, Hubert & Li, Fred & Liang, Xiaohui. (2023). A Mixed Reality Training System for Hand-Object Interaction in Simulated Microgravity Environments. 167-176. 10.1109/ISMAR59233.2023.00031.

Medeiros, A.C.S., Ratsamee, P., Uranishi, Y., Mashita, T., Takemura, H. (2020). Human-Drone Interaction: Using Pointing Gesture to Define a Target Object. In: Kurosu, M. (eds) Human-Computer Interaction. Multimodal and Natural Interaction. HCII 2020. Lecture Notes in Computer Science(), vol 12182. Springer, Cham. <https://doi.org/10.1007/978-3-030-49062-1_48>

Zhao, Z. Q., Zheng, P., Xu, S. T., & Wu, X. (2019). Object detection with deep learning: A review. IEEE transactions on neural networks and learning systems, 30(11), 3212-3232.

Laroca, R., Severo, E., Zanlorensi, L. A., Oliveira, L. S., Gonçalves, G. R., Schwartz, W. R., & Menotti, D. (2018, July). A robust realtime automatic license plate recognition based on the YOLO detector. In 2018 International Joint Conference on Neural Networks (IJCNN) (pp. 1-10). IEEE.

Huang, R., Pedoeem, J., & Chen, C. (2018, December). YOLO-LITE: a real-time object detection algorithm optimized for non-GPU computers. In 2018 IEEE International Conference on Big Data (Big Data) (pp. 2503-2510). IEEE.

Xie, P., Chen, S., Hu, D., Dai, Y., Yang, K., & Wang, G. (2024). Target-Oriented Object Grasping via Multimodal Human Guidance. *arXiv preprint arXiv:2408.11138*.

Medeiros, A.C.S., Ratsamee, P., Orlosky, J. *et al.* 3D pointing gestures as target selection tools: guiding monocular UAVs during window selection in an outdoor environment. *Robomech J* **8**, 14 (2021). https://doi.org/10.1186/s40648-021-00200-w

Xu, J., Wang, H., Zhang, J., & Cai, L. (2022). Robust hand gesture recognition based on RGB-D Data for natural human–computer interaction. *IEEE Access*, *10*, 54549-54562.

Chang, V., Eniola, R. O., Golightly, L., & Xu, Q. A. (2023). An Exploration into Human–Computer Interaction: Hand Gesture Recognition Management in a Challenging Environment. *SN Computer Science*, *4*(5), 441.

Pelgrim, M. H., He, I. X., Lee, K., Pabari, F., Tellex, S., Nguyen, T., & Buchsbaum, D. (2024). Find it like a dog: Using Gesture to Improve Object Search. In *Proceedings of the Annual Meeting of the Cognitive Science Society* (Vol. 46).

Jirak, D., Biertimpel, D., Kerzel, M., & Wermter, S. (2021). Solving visual object ambiguities when pointing: an unsupervised learning approach. *Neural Computing and Applications*, *33*, 2297-2319.

Etoh, M., Tomono, A., & Kobayashi, Y. (1989). Direct Finger Pointing as a Man-machine Interface. *IFAC Proceedings Volumes*, *22*(12), 125-130.

Lee, K. (2024). *Point it Out: Using Gesture to Improve Object Search* (Doctoral dissertation, Brown University Providence, Rhode Island).

Wong, N., & Gutwin, C. (2010, April). Where are you pointing? The accuracy of deictic pointing in CVEs. In *Proceedings of the sigchi conference on human factors in computing systems* (pp. 1029-1038).

Rümelin, S., Marouane, C., & Butz, A. (2013, October). Free-hand pointing for identification and interaction with distant objects. In *Proceedings of the 5th International Conference on Automotive User Interfaces and Interactive Vehicular Applications* (pp. 40-47).

Aftab, A. R., von der Beeck, M., & Feld, M. (2020, October). You have a point there: object selection inside an automobile using gaze, head pose and finger pointing. In *Proceedings of the 2020 International Conference on Multimodal Interaction* (pp. 595-603).

Florian Roider and Tom Gross. 2018. I See Your Point: Integrating Gaze to Enhance Pointing Gesture Accuracy While Driving. In Proceedings of the 10th International Conference on Automotive User Interfaces and Interactive Vehicular Applications. 351–358.

Harika, M. (2016). *Finger-pointing gesture analysis for slide presentation* (Doctoral dissertation, 부경대학교 대학원).

Erden, F., & Cetin, A. E. (2014). Hand gesture based remote control system using infrared sensors and a camera. *IEEE Transactions on Consumer Electronics*, *60*(4), 675-680.

Erden, F., Velipasalar, S., Alkar, A. Z., & Cetin, A. E. (2016). Sensors in assisted living: A survey of signal and image processing methods. *IEEE Signal Processing Magazine*, *33*(2), 36-44.

Sudhakar, S., Liu, R., Van Hoorick, B., Vondrick, C., & Zemel, R. (2024). Controlling the World by Sleight of Hand. *arXiv preprint arXiv:2408.07147*.

Nelson A, McCombe Waller S, Robucci R, Patel C, Banerjee N. Evaluating touchless capacitive gesture recognition as an assistive device for upper extremity mobility impairment. Journal of Rehabilitation and Assistive Technologies Engineering. 2018;5. doi:10.1177/2055668318762063

Jing, L., Cheng, Z., Zhou, Y., Wang, J., & Huang, T. (2013, December). Magic ring: A self-contained gesture input device on finger. In *Proceedings of the 12th International Conference on Mobile and Ubiquitous Multimedia* (pp. 1-4).

Inomata, S., Komiya, K., Iwase, K., & Nakajima, T. AR Smart Home: a Smart Appliance Controller Using Augmented Reality Technology and a Gesture Recognizer.

Tanada, K., Matsuzaki, S., Tanaka, K., Nakaoka, S., Kondo, Y., & Mori, Y. Pointing Gesture Understanding via Visual Prompting and Visual Question Answering for Interactive Robot Navigation. In *First Workshop on Vision-Language Models for Navigation and Manipulation at ICRA 2024*.

# Appendix A

1. The actual device control research and implementation are not in the scope of this praxis. [↑](#footnote-ref-1)