THE UNIVERSITY OF TEXAS AT ARLINGTON

DOCTORAL DISSERTATION

Computer Vision Methods for Sign Language Recognition and Cognitive Evaluation through Physical Tasks

---

Author:
Alex DILLHOFF

Supervisor:
Dr. Vassilis ATHITSOS

Presented to the Faculty of the Graduate School of The University of Texas at Arlington in Partial Fulfillment of the Requirements for the Degree of

DOCTOR OF PHILOSOPHY

August 4, 2020

Declaration of Authorship

I, Alex DILLHOFF, declare that this dissertation titled, "Computer Vision Methods for Sign Language Recognition and Cognitive Evaluation through Physical Tasks" and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at The University of Texas at Arlington.
- Where any part of this dissertation has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this dissertation is entirely my own work.
- I have acknowledged all main sources of help.
- Where the dissertation is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed:

Date:

“*If we understood the world, we would realize that there is a logic of harmony underlying its manifold apparent dissonances.*”

Jean Sibelius

THE UNIVERSITY OF TEXAS AT ARLINGTON

Abstract

Faculty Name

Computer Science & Engineering

Doctor of Philosophy

Computer Vision Methods for Sign Language Recognition and Cognitive Evaluation through Physical Tasks

by Alex Dillhoff

Analyzing human motion is vital for a multitude of tasks including human-computer interaction, sign language recognition, and the assessment of cognitive disorders. Providing automatic assessments for cognitive disorders increases the accessibility and affordability of life-changing tests and treatments. For sign language recognition, automated translation systems bridge the gap between native and non-native signers. Additionally, dictionary look-up systems are helpful for native signers learning a new language. Common to both of these tasks is the reliance of fine motor function in the hands. Hand Pose Estimation methods are used to drive applications that rely on hand shape. These tasks present unique and difficult challenges which are investigated in this dissertation.

We present our preliminary data analysis towards an automated assessment system for the Activate Test of Embodied Cognition (ATEC), a measurement of cognitive skills through physical activity. Evaluating cognitive function through physical movement requires data from many participants performing a wide variety of physical tasks. Collecting such a dataset is a time-consuming yet worthwhile goal. Automatically scoring the movements of each task requires that the method be robust to noise as well as accurate to ensure proper recommendations are made to experts. We evaluate three ATEC tasks designed to address attention, working memory, response inhibition, rhythm and coordination in children: *Sailor Step*, *Ball-Drop-to-the-Beat*, and *Finger Tap*. These tasks are specifically designed to assess lower and body accuracy, response inhibition, rhythm, and gross motor function. We present our data collection framework and evaluate baseline methods on real ATEC data.

In sign languages, a periodic sign is one that contains repeated movements. Dynamic Time Warping (DTW) is often used in sign language recognition to generate a frame alignment between two input signs that provides a measure of their similarity. Alignments provided by DTW may be erroneous when the input contains periodic signs, especially when the number of periods differs between inputs. Additionally, the number of periods may change between individual signers and signs. Little work has been done to address the problem of recognizing periodic signs in the context of DTW. This work evaluates two DTW-based approaches. The first uses a newly defined periodic warping path. The second uses manual annotations to truncate periodic input to contain no more than two periods. These two methods are compared against a standard implementation of DTW. Recognition accuracy and quality of alignment are analyzed. The results motivate a need for further research in periodic sign language recognition.

Deep learning based 3D hand pose estimation requires large amounts of data for training. Fully supervised methods provide reasonable accuracy but require 3D annotations for each individual frame in the dataset. Providing such annotations is an expensive task that may be infeasible for many novel applications. This dissertation investigates self-supervised methods for 3D hand pose estimation models with little or no joint annotations.

The self-supervised component is based on a 3D hand model and will generate a sample of the predicted pose. Parameter choices are evaluated to determine the best representation for hand pose and shape. The predicted hand shape is compared with the input depth image as a means of supervision which can be used in parallel with joint annotations. We consider two cases: one in which limited annotations exists as well as when the model is trained with only unlabelled samples.

Acknowledgements

I am fortunate to have good people in my life who want to see me succeed. I love them all and am thankful for every day that I know them. I would like to give a special thank you to my advisor, Vassilis Athitsos, who helped me realize that I wanted to pursue this degree in the first place. His support and advice have been invaluable over these past years and have surely contributed to my success. Vassilis provided the perfect balance in which I could explore topics that interested me while maintaining focus on the realities of deadlines. I thank Chris Conly, my office-mate turned committee member. I could not have asked for a better mentor and friend to guide me into the world of graduate education. Thank you to Farhad Kamangar for answering my many questions throughout the years. You challenged me to question my understanding of all things and convinced me that hand-wavy answers are never satisfying. To Dave Levine, thank you for always showing confidence in me. Your assuredness in my abilities was vital in removing whatever doubts I may have had at the time.

I am very thankful for discussions I had and topics I learned from the members of the VLM lab. Doing anything alone is the worst way to reach a goal. To Saif Sayed, Srujana Gattupalli, Sakher Ghanem, Amir Ghaderi, Reza Ghoddoosian, Marnim Galib, Farnaz Farahanipad, and many others, I thank you. You are all incredible and kind people who deserve nothing but success. I am rooting for everyone and will always be here for whatever you may need.

I would also like to thank Morris D. Bell and Fillia Makedon. I am very fortunate to have been included in such an incredible and meaningful project that changed the outcome of this dissertation for the better. The opportunities for inter-disciplinary collaboration have made me a better person in many ways. I hope to continue working on this impactful project as well any future endeavours you may have.

Contents

List of Figures

List of Tables

For my father, Doug, thank you for everything. To Ed, my grandfather, I have always cherished our time together and cannot wait for our next project. To my love, Kalee, you never stopped encouraging my growth. I will spend the rest of my life repaying you all.

Chapter 1

Introduction

Problems in human motion analysis are challenging because of the many degrees of variance present in each motion. There are many subtasks of motion analysis including the identification of body parts, recognizing specific actions, tracking individual movement, and the evaluation of mental and physical health. Visual data for each task can vary in perspective, resolution, and modality. Computer vision and machine learning solutions must be able to account for such variance without sacrificing performance. Fully supervised models for such tasks require manual annotations for each sample in the dataset. This requirement is cumbersome at best and prohibitive at worst. This dissertation proposes solutions for gesture recognition, activity recognition, and hand pose estimation. Additionally, the broader challenge of reducing the burden of manual annotations for training machine learning models is discussed. A method is proposed to bridge the gap of model performance that comes with a reduced amount of annotated data.

This dissertation considers several problems related to evaluating physical movement in video. Specifically, a method for isolated sign language recognition is proposed which considers a special class of signs. A dataset for evaluating cognitive ability through physical exercise is introduced along with baseline approaches for automatically scoring selected tasks. From these tasks arise additional important problems, especially when considering novel datasets and environments. In this chapter, the primary problems investigated in this dissertation are briefly reviewed followed by a summary of contributions.

1.1 Contributions

This dissertation makes contributions towards Automatic Sign Language Recognition, Automatic Cognitive Evaluation through Physical Tasks, and Hand Pose Estimation using self-supervised methods. This section highlights these contributions as the pertain to each topic.

1.1.1 Recognition of Periodic Signs

Automatic Sign Language Recognition is the task of translating a video sequence of physical motions to the corresponding sign. There are two types of this task to consider – isolated and continuous. For isolated sign language recognition, a video of isolated motion must be translated into a single sign. Continuous sign language recognition is a more challenging tasks which considers a long sequence of video, typically depicting a full sentence. The increased challenge is due to the fact that the individual signs themselves are not segmented. Thus, any proposed method for continuous sign language recognition must also consider meaningless frames between signs.

This dissertation examines a special case of signs that are periodic in nature. The physical motions present in any individual sign exhibit several types of variance due to the physical size of the signer, camera perspective, emphasis, context, and more. Further, there are certain signs which have repeated, or periodic, motions. A system which does not consider periodic motions would erroneously translate two iterations of the same sign into different words, simply because one of the sequences included a repeated motion.

There are several datasets available for sign language recognition, however they are not generally cross-compatible as there is no universal sign language. Given the smaller number of examples per sign available, the classification approach proposed in this dissertation compares input and database signs using a table lookup method. By modifying the constraints of Dynamic Time Warping, we are able to correctly classify signs with varying degrees of periodicity. We compare our results with a standard implementation of

Dynamic Time Warping, leading to an overall improvement in performance on a publicly available sign language dataset.

1.1.2 Automatically Evaluating Cognitive Performance through Physical Tasks

Attention Deficit Hyperactivity Disorder (ADHD) is a neuro-developmental disorder observed in children as early as age 6. Given the overlap of its symptoms with other disorders, it is difficult to diagnose. Underdevelopment of certain cognitive abilities can affect the learning outcomes of children. Left unmanaged and untreated, the symptoms can affect individuals well into adulthood, therefore, early diagnosis of cognitive problems is paramount as brain plasticity diminishes with age. Thus, proper diagnosis and treatment options are vital for successful outcomes.

Toward providing accessible and affordable tools for the diagnosis and treatment of cognitive disorders such as ADHD, this dissertation introduces a novel dataset of children performing selected physical tasks that examine specific Executive Functions from the Activate Test of Embodied Cognition (ATEC). This data is critical in the development of automatic methods for scoring individual tasks. Additionally, analysis of the data can provide insight into how physical exercise can be used to train and evaluate cognitive skills.

We present the details of our data capture framework that is used for recording child participants performing ATEC tasks. Further, we present the results of our baseline evaluations on three selected tasks: *Bag Pass*, *Sailor Song*, and *Finger Tap*.

1.1.3 Self-Supervision for Hand Pose Estimation

Hand Pose Estimation is the task of predicting pose parameters of the hand including joint keypoints and physical hand shape and is useful in a wide array of applications such as Virtual Reality, Human-Computing Interfaces, and physical rehabilitation. In the context of this dissertation, it is important when considering both Sign Language Recognition and specific ATEC tasks. For Sign Language Recognition, many signs that share similar hand motion differ only in the shape of the hand. Without any knowledge of hand shape, a sign language recognition method would confuse two completely different signs that shared a similar hand trajectory. The *Finger Tap* and *Finger Appose Succession* tasks of the ATEC suite must properly evaluate the hand shape in order to successfully score participants.

Adapting state-of-the-art hand pose estimation methods toward realistic and novel datasets, such as ATEC, is challenging due to the unique environments in which they are captured. Further, no standard hand pose estimation dataset considers samples of a child's hand. Manually annotating the 3D pose and shape of the hand is a laborious and error-prone task.

This dissertation evaluates a method for self-supervised hand pose estimation as the number of annotated samples is reduced. The self-supervised approach discussed decouples the dependence on how the joint and shape parameters are estimated and the way in which samples are generated. Formulating the problem in this way allows the method to utilize the latest research in Computer Vision and Machine Learning.

1.2 Dissertation Structure

Each chapter presented in this dissertation is meant to be self-contained, with all relevant background material provided as necessary. Chapter 2 presents the proposed method of identifying periodic signs, which we refer to as Periodic Dynamic Time Warping (PDTW).

Chapter 3 describes the Activate Test of Embodied Cognition and its broader societal impact. The data capture framework, which is used in the collection of all ATEC-related data, is described. Three of the ATEC tasks are defined and baseline evaluations are considered.

Chapter 4 presents the work towards self-supervised hand pose estimation. This is a challenging task for which a working method would have impactful outcomes for the tasks described in previous chapters. Background of Hand Pose Estimation is provided and some preliminary results are discussed.

This dissertation concludes with a discussion on the key components of future research towards a fully automated scoring approach for ATEC. Providing such automatic solutions would have impacts for both childhood cognitive development and machine learning research.

Chapter 2

Cognitive Evaluation through Physical Activity

Executive functions are high-order cognitive processes involved in multitasking, time management, attention, planning, inhibition, self-regulation and memory. They typically develop well into a person's third decade of life. Children with Attention-Deficit Hyperactivity Disorder (ADHD) exhibit weaknesses in executive functions, specifically response inhibition, planning, vigilance, and working memory [81]. These weaknesses manifest themselves both physically and mentally. Examples include:

- inability to focus for long periods time,
- unable to follow instructions,
- avoiding tasks that require mental effort,
- difficulties with organization,
- moving, running, or climbing in a way that is not appropriate,
- inability to sit still,
- blurting out responses in a distracting manner,
- and inability to consider important, long-term decisions.

Cognitive impairments in early childhood can lead to poor academic performance and require proper assessment and intervention at the appropriate time [43]. Such impairments can last well into adulthood, resulting in lower high school graduation rates [6, 5], poor job performance [6], and lower GPAs in college [29]. Additionally, higher rates of alcohol and other substance abuse has been reported [44].

Attentional Control

Choosing to pay attention to one thing over another.

Cognitive Inhibition

Ability to tune out irrelevant information.

Inhibitory Control

Override natural urges and responses.

Working Memory

Manipulation of active memory (think L1 cache).

Cognitive Flexibility

Ability to switch between tasks effectively.

Physical activities are an important manifestation of cognitive functions [21]. Not only can they be utilized to assess cognitive skills [14], but they can also be used to train such skills [75]. Many of these activities are simple enough to be implemented in a group setting with little to no additional equipment, making classrooms an ideal environment for training. An important challenge is how to measure the performance of each physical activity, for which there is currently little understanding.

Assessment and diagnosis of cognitive disabilities such as attention deficit disorders can be performed by a variety of experts including psychiatrists, pediatricians, and social workers. Not all experts specialize in the diagnosis and treatment of attention deficit disorders, which can have a varied effect on the outcomes of patients. For many individuals, access to an expert for even diagnosis can be prohibitive due to financial reasons, let alone long-term care and treatment. A review of studies covering the economic impacts of ADHD estimated a national annual cost range of $143 to $266 billion [22]. Russell et al. found that children from disadvantaged socioeconomic backgrounds are at an increased risk of having ADHD [60]. The economic and personal impacts of ADHD is high and calls for cost-effective diagnostic tools and treatments. Automating the performance of key physical tasks would provide several benefits including:

- easier access to evaluations for cognitive disabilities,
- reduced requirement on expertise for health care professionals,
- and training for children during key developmental stages.

Although providing measurements for executive function through physical activity is relatively unexplored, there have been other attempts to do so using non-physical computer-based assessments. First, a brief overview of automatic EF evaluation provides further context and motivation for the development of computer vision-based methods.

In section 2.2, a novel framework for measuring executive function in children through physical tasks is described. In subsequent sections, we present our proposed methods to automatically administer and assess two core ATEC tasks: the Ball Drop task and Sailor Step task. These tasks were designed to assess upper-body (hands) and lower-body (feet) movements. We describe the two tasks, as well as the experimental approach towards an automated scoring system through computer vision and machine learning methods. These methods are trained and evaluated on the dataset described in Section 2.3, consisting of child participants between 6 and 10 years old. Preliminary results for both tasks indicate the efficiency of our proposed methods towards an automated assessment system for embodied cognition in children.

2.1 Background on Evaluating Executive Function

The NIH toolbox, a standardized test used for cognitive assessment [84] and other existing computer-based assessments are extensively used to assess executive functions in children, but they require little body movement and may be less closely related to assessing cognition in motion than daily functioning.

Providing a system for automatic assessment can provide more opportunities for diagnosis, treatment, and the progress of cognitive skills. Physically active behaviors are important in the daily lives of children and have implications for fitness, learning, social interactions, and physical and psychological development [41]. Moreover, studies have shown a measured improvement in cognitive skills and academic performance in children associated with increased physical fitness [14, 21]. These indicate the strong relation between motor and cognitive development in children and their implications to daily functioning [15]. While there are existing assessment systems for both motor development [72] and neurocognitive measures (NIH) [84], as well as for assessing emotional and behavioral problems (CBCL) [2] and executive function behaviors at home and at school (BRIEF) [66], these are either computer-based or paper-based in the form of parent/teacher reports and require no movement.

Our proposed embodied cognition assessment system, evaluated on ATEC data, utilized the advances of computer vision and machine learning methods to analyze a child's performance during a set of physical tasks specifically designed to extract information about executive function, motor function, and development.

Action recognition methods use image or body key-point data to model the spatial and temporal features of each class-action [36, 18, 3]. These methods are the most suitable for our solution as all ATEC tasks involve physical movement. Action recognition often involves classifying high-level events with more variation between classes [42, 63, 35]. Zhang et al. report that many image-based action recognition datasets feature low variability amongst actions [85]. Although each individual ATEC task must be performed in a specific manner, there can be large variations between each individual's performance. As such, our approaches to these tasks must follow other methods and datasets with high intra-class variance [53, 26, 48]. The body key-points are the most salient high-level features for these tasks. Recent body pose estimation methods have shown reasonable results on benchmarks featuring multiple persons with varying viewpoints and lighting [9, 25, 52]. Since our participants are relatively close to the cameras and are recorded with good lighting, we are able to get high quality key-point estimates.

We use the DeepGRU [38] sequence model as a benchmark for the *Ball Drop* and *Sailor Song* tasks since it requires fewer parameters than other recurrent models. The authors show good performance even with smaller datasets which is especially important in the early stages of dataset development, when we have a relatively small number of examples per class. The goal of our work is to develop efficient and reliable methods for child activity recognition, since detecting and analyzing movement performed by children is challenging due to high variability in performance and a large amount of random movements.

2.2 The Activate Test of Embodied Cognition

The Activate Test of Embodied Cognition (ATEC) is an assessment test designed to measure executive functions in children through physically and cognitively demanding tasks and provides measurements for attention, working memory, response inhibition, self-regulation, rhythm and coordination, as well as motor speed and balance.

ATEC consists of 17 physical exercises with different variations and difficulty levels. It is designed to provide measurements of executive and motor function, including sustained attention, self-regulation, working memory, response inhibition, rhythm, and coordination, as well as motor speed and balance [TODO: literally just stated this]. These measurements are converted to a final ATEC score which describes the level of development (e.g., early, middle, full development).

- Gross Motor – Gait and Balance
  - Walk Forward – Baseline measurement of subject's gait.
  - Gait on Toes – Subject walks on the balls of the feet.
  - Tandem Gait – Subject walks heel-to-toe.
  - Standing with Arms Outstretched – Subject should keep arms perpendicular to body.
  - Stand on One Foot (left/right) – Measurement of balance.
- Rhythmic Movement
  - March to the Beat – Taking a marching step in rhythm with a steady beat.
- Bilateral Coordination

  - Ball Pass – Pass the ball from one hand to the other.

- Auditory/Visual Upper Body Accuracy, Rhythm, and Response Inhibition
  - Ball Pass Red/Green Light – Ball passing task with go/no-go cues.
  - Ball Pass Red/Green/Yellow Light – Additional cue and action during task.
- Visual Lower Body Accuracy, Rhythm, and Response Inhibition
  - Sailor Song – Rhythmic movements in coordination with a song.
- Bilateral Coordination and Self-Regulation
  - Ears, Shoulders, Hips, and Knees – Touch the body part corresponding to the instruction. Subsequent trials increase the difficulty by re-assigning each instruction (e.g. Shoulder means Knees).
- Chin, Nose, Lips, and Forehead
  - Right hand – Touch a specific point of the face starting with an outstretched right hand.
  - Left hand – Same as before with opposite hand.
  - Both hands – Using both hands at the same time.
- Rapid/Sequential Movements
  - Foot Tap – Tap the toes of the feet as fast as possible for 10 seconds.
  - Heel-to-Toe Tap – Alternate between heel and toe tapping.
  - Hand Pat – Tap the hand to the leg as fast as possible for 10 seconds.
  - Hand Pronate/Supinate – Alternate between the back of the hand and palm.
  - Finger Tap – Tap the index and thumb together as fast as possible for 10 seconds.
  - Appose Finger Succession – Tap the thumb and each finger together in succession.

Besides the development of methods for the automatic evaluation of ATEC tasks, there are several key goals that this research aims to achieve. Through the analysis of the data collected during this research, we can identify important correlations between physical performance and cognitive function. We also gain insight into long-term development as subjects participate in follow-up evaluations. An important application of this research is to design a high-fidelity and low-cost automated assessment system which analyzes the movements of the performed tasks and produces reliable cognitive measures. Such a cost-effective measure could increase the availability of diagnosis and treatment options for more children, especially those from disadvantaged socioeconomic backgrounds.

2.3 Data Collection Framework

The variation in which each ATEC task can be performed between subjects means a large sample size is required to develop automatic methods for evaluation. To ensure that such variation can be captured, multiple viewpoints and modalities are used. We introduce a novel dataset for computational cognitive assessment through physical tasks. This section reviews the recording protocol and development of the framework used for research related to ATEC.

This dataset is an important requirement for the development of human motion analysis algorithms and automated scoring methods which compute various metrics of physical performance. Discovery of new knowledge related to physical exercises in cognitive training and correlations between individual metrics can be achieved by analyzing extraneous and unforeseen movements within the data. Further, the data will be used to develop a system for cognitive assessment which tracks the progress of a child as they complete cognitive assessments to monitor improvements over time and provide recommendations and decision support for cognitive experts.

We record the participants performing each of the 40 tasks as part of the ATEC battery of tests. Participating children record an initial session as well as a follow-up session two weeks later. To our knowledge this is the first large-scale dataset of physical tasks related to measuring embodied cognition.

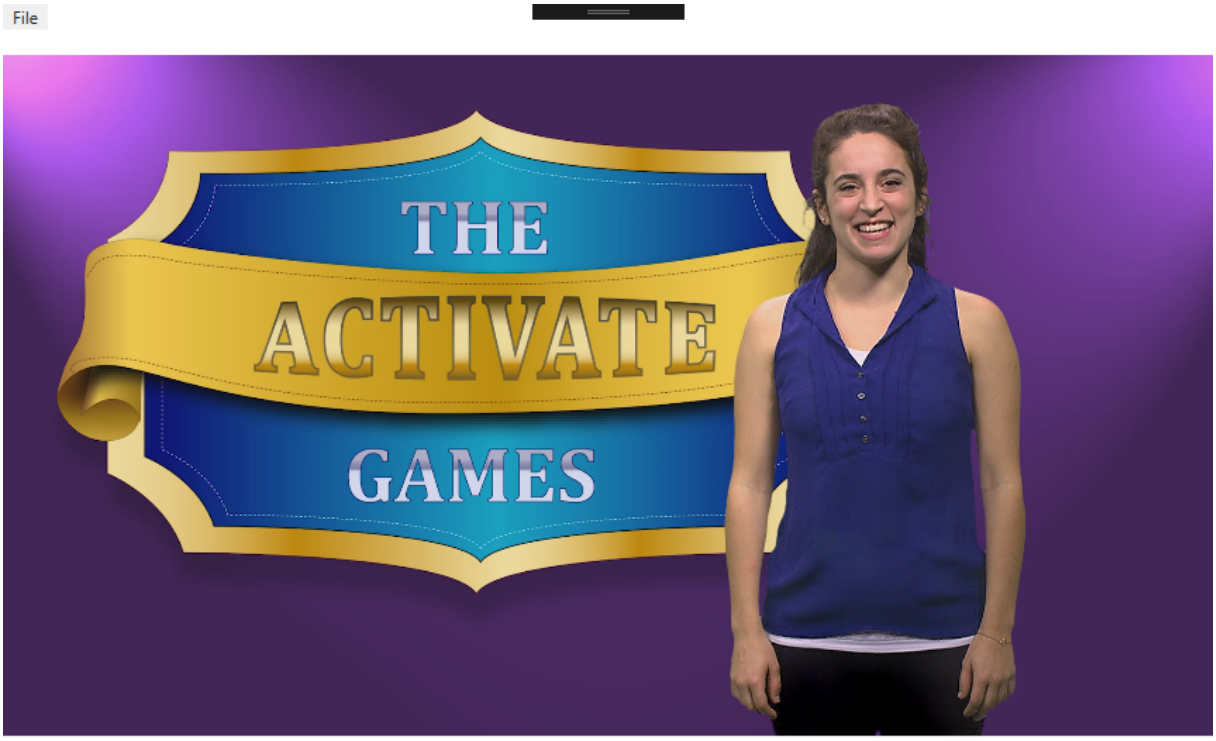


FIGURE 2.1: The ATEC battery of tests are guided by the instruction of Aliza. These videos play within the recording application.

A large amount of data is required to study correlations between performance in physical exercises and the level of specific cognitive skills. Through regular training and assessment, improvement of these skills can also be observed. To the best of our knowledge, there exists no automatic method for assessing individual performance during physical exercises.

2.3.1 Technical Specifications - How is the data recorded?

Besides ensuring that the maximum amount of useful data is recorded for each session, several other factors were considered during the development of this dataset. It is important that the data collection not be distracting as the child participant performs a series of unique tasks. Further, administration of the task must be simple and robust. We developed a recording framework and protocol which embodies both of these points during each recording session.

We use two Microsoft Kinect V2 cameras to record RGB, depth, audio, and body

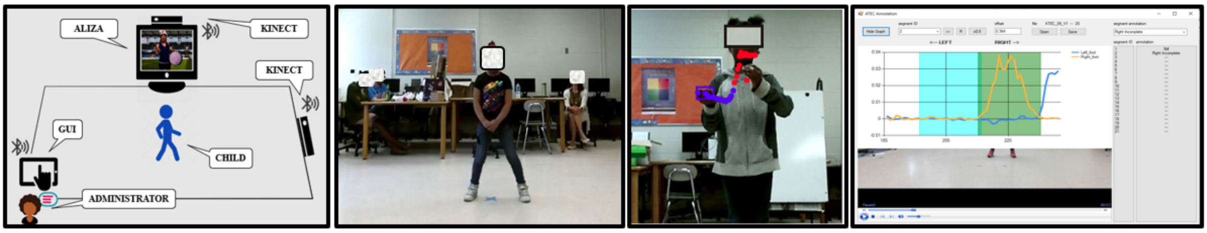
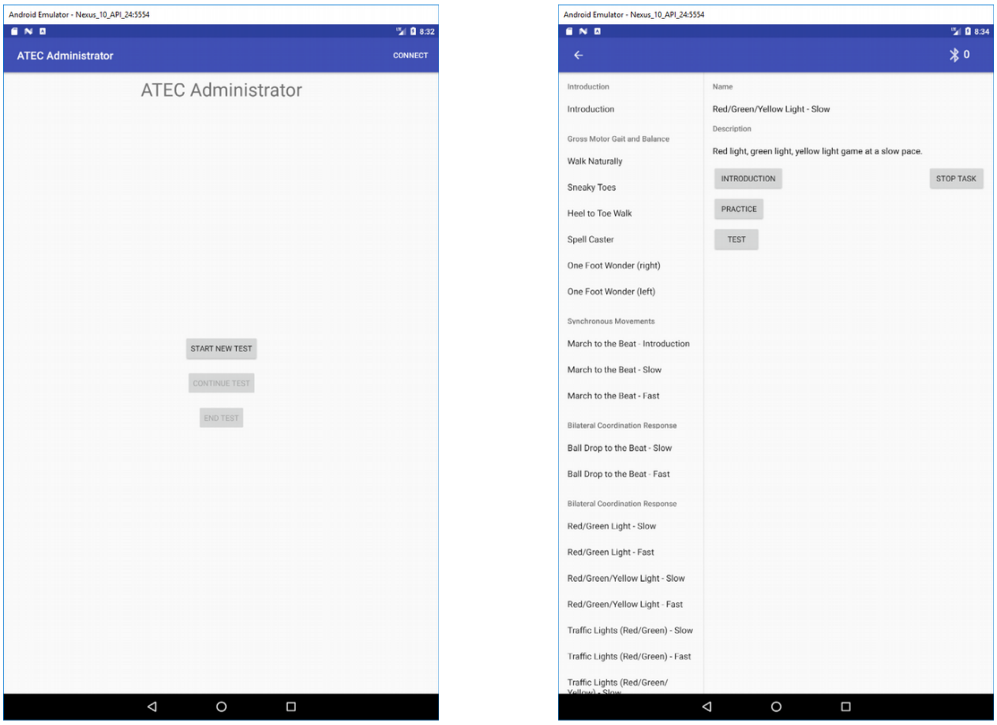


FIGURE 2.2: The ATEC setup includes two Kinect cameras, a large screen and a tablet interface for the administrator. Administration takes place in classroom environments. Annotation software was developed to enhance manual scoring and annotate the collected data, given the task rules and the cognitive measures to be assessed.

keypoints. The cameras are positioned such that both a front and side view of the participant is captured. This is especially necessary for tasks which involve gait analysis as computer vision methods may be less effective using only a frontal view. The Microsoft Kinect V2 camera record color video at 30 frames per second with a resolution of 1920 × 1200. The depth sensor records at 30 frames per second with a lower resolution of 512 × 424. All data is recorded in a lossless manner using a custom file format tailored to accommodate multiple sources of multi-modal data.

The administrator controls the progress of the test battery through a tablet application written for the Android operating system. Figure 2.3 shows two screen shots of the final application. This interface allows the administrator to start and stop tests, monitor camera performance, and log tasks as they are performed. Monitoring recording performance is critical to ensure that each task is captured without frame drops. Completing the test battery takes about 45 minutes and repeating tasks should be avoided.

The recording software is developed using the Microsoft .NET framework, which was necessary to utilize the Kinect API natively. Since the recording protocol requires two cameras, the software supports multiple systems. The administrator application communicates with each recording PC via Bluetooth. Figure 2.2 shows the layout of the recording space along with visual samples of the participants, tracking, and annotation tool.


Android Emulator - Nexus_10_API_24:5554
8:32
ATEC Administrator
CONNECT
ATEC Administrator
START NEW TEST
CONTINUE TEST
END TEST
Android Emulator - Nexus_10_API_24:5554
8:34
0
Introduction
Introduction
Gross Motor Gait and Balance
Walk Naturally
Sneaky Toes
Heel to Toe Walk
Spell Caster
One Foot Wonder (right)
One Foot Wonder (left)
Synchronous Movements
March to the Beat - Introduction
March to the Beat - Slow
March to the Beat - Fast
Bilateral Coordination Response
Ball Drop to the Beat - Slow
Ball Drop to the Beat - Fast
Bilateral Coordination Response
Red/Green Light - Slow
Red/Green Light - Fast
Red/Green/Yellow Light - Slow
Red/Green/Yellow Light - Fast
Traffic Lights (Red/Green) - Slow
Traffic Lights (Red/Green) - Fast
Traffic Lights (Red/Green/Yellow) - Slow
Name
Red/Green/Yellow Light - Slow
Description
INTRODUCTION
STOP TASK
PRACTICE
TEST

Red light, green light, yellow light game at a slow pace.

(A) The main screen of the administration application.

(B) List of tasks within administration application.

FIGURE 2.3: ATEC Administration Android Application

2.3.2 Dataset Details

Including the variations on these 17 physical exercises, there are 40 individual tasks that are recorded in each session with each task being around 15 seconds. On average there are close to 72,000 combined RGB and depth frames recorded over a single session. A full length assessment takes about 45 minutes including pauses between tasks.

Data collection is ongoing and we currently have over 50 subjects recorded with 2 sessions per subject. The initial session establishes a baseline for each subject. A followup session is scheduled 2 weeks after the initial one and includes the exact same sequence of tasks.

Each task is annotated with two different types of annotations. For behavior evaluation, psychiatric experts annotate each task following the related ATEC scoring criteria. For automatic scoring through computer vision and machine learning, annotations are based on the task. In most cases, the videos are assigned sequence-based annotations as the task is treated as an activity recognition problem. However, certain tasks require frame-level annotations.

2.4 Ball Drop Task

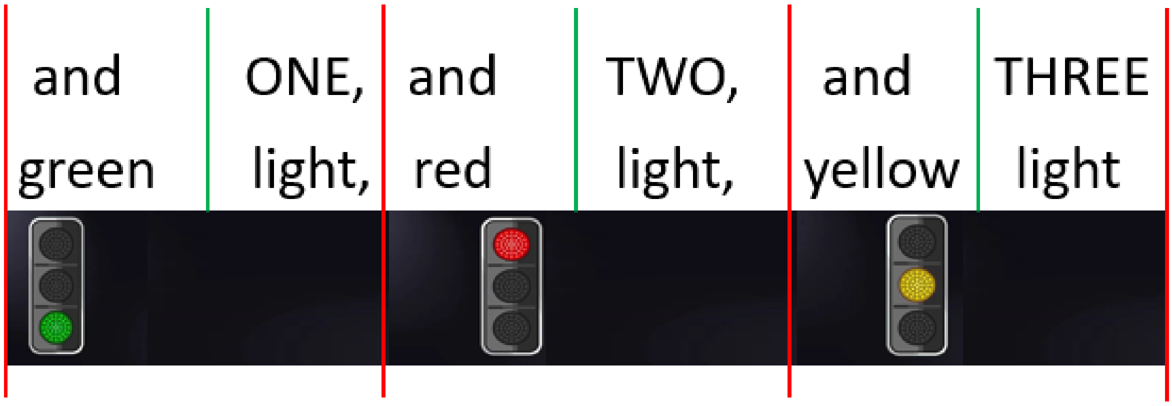


Figure 2.4: Audiovisual stimuli during the Ball Drop task. Each segment requires a specific activity (red lines). For the audio tasks, each segment includes two beats (green line).

Ball Drop is a core ATEC task designed to assess response inhibition, lower-body movement accuracy, and rhythm. It additionally assesses both audio and visual cue processing. The participant is required to pass a ball from one hand to another, following audio and visual instructions. The task modifies the rules of the Red-Light/Green-Light game in which the participants are required to perform certain movements while they hold a ball. Based on the rules, the child is instructed to pass the ball on a green light, keep the ball still for a red light, and move the ball up and down with the same hand for a yellow light. The light colors are presented both audibly and visually to measure both audio and visual accuracy and response inhibition. The task is assessed at 60 beats per minute for the slow version and 100 beats per minute for the fast iteration.

During this assessment, the stimuli are presented as pictures of traffic lights: red, green, and yellow. The child is instructed to do the appropriate movement with the ball when a new picture of a traffic light appears. Apart from accuracy and response inhibition, this exercise also assesses rhythm. The ATEC on-screen host, Aliza, presents the stimuli in a rhythmic manner by saying "green/red/yellow light" in two beats; one for the color word and one for the word "light." The children are instructed to perform the movements in two beats. For pass and raise commands, the ball is raised on the first beat and either passed or lowered on the second. To acclimate the participants to the task, they are instructed to pass the ball eight times following the rhythm of the spoken instructions. Figure 2.5 visualizes both the audio and visual stimuli used in this task.

The automated scoring produced by our method must match the scoring system as defined by psychiatric experts. The scoring is split between three categories with both a raw and converted value. The converted values of each category over all task variations are within the range $[0, 8]$. A visual accuracy score reflects the number of correct passes taken, independent of rhythm. The visual response inhibition score depicts the participant's ability to stay still when prompted with a red light. There are 12 cues per task that measure response inhibition. For rhythm, a total of 32 raw points can be earned. Since each prompt consists of two beats, partial points can be given. For example, a single point is given if the hand is raised on beat, but the passing movement is not.

2.4.1 Automated Scoring Pipeline

For this task, there are three main cues involved: ball pass, no ball pass, and hand raise. The participant is prompted to perform one of the actions corresponding to a visual or auditory cue, depending on the variant of the task described previously. The proposed automated system must detect the actions performed and score them according to the rules defined by domain experts. More specifically, the system must include accuracy, response inhibition, and rhythm scores.

Given an input video $\mathbf{X} \in \mathbb{R}^F$ consisting of F frames, subsequences are extracted based on the timestamps of each visual and auditory cue. In the base visual response task, there are 10 passing cues and 12 red light cues. For the green/yellow/red version, there are 12 total passing cues and 8 red light cues. The frames corresponding to each cue are segmented so that the input represents the participant's response to a single command.

We evaluate two different approaches towards automatically scoring the *Ball Pass* task. As the goal is to provide scores that match those provided by expert reviewers, it is imperative that any proposed methods be accurate and robust to noisy input. The first method uses a popular object detection framework combined with Dynamic Programming. This implementation was used on a small initial sample of the dataset during the earlier stages of data collection, at which point there were only 6 annotated subjects from their initial evaluation on only the first *Ball Drop* related task. We also evaluate a deep learning approach using a successful recurrent model using data from 7 subjects performing 4 versions of the *Ball Drop* task across 2 sessions each. In total 56 recorded tasks were considered.

Dynamic Programming Method

Our first approach first detects the ball in each frame using YOLOv3 [58], a popular object detection model. This was chosen specifically for its fast inference time and relatively high accuracy when detecting smaller objects, such as a ball held in a child's hand. Toward the goal of creating a robust method, we use YOLOv3 to produce multiple object candidates per frame. We found that our model would provide erroneous detections. This is due to the small resolution of the ball captured from each frame. In order to remain robust to these types of detections, we use a Dynamic Programming approach to calculate the most likely path of detections over time. Any remaining outliers will be ignored based on the chosen cost function.

To train our model, we annotate the bounding boxes of the ball from the processed ATEC data over 6 different subjects. The raw RGB frames were cropped to a size of 416×416, centered on the torso using the keypoints provided by the Kinect V2 camera. In total, there are 5168 annotated frames for training. One user-independent video consisting of 356 frames is set aside for testing.

We use a Viterbi-based Dynamic Programming algorithm to calculate the most likely path given multiple candidates per frame. The method is defined by:

- A **cost matrix** $T_{cost}[i, j]$ which stores the cost of the most likely path observed so far.

- A **path matrix** $T_{path}[i, j]$ which stores the state of the most likely path ending at state s_{j-1}.
- A **transition cost** A_{ij} which is the cost to transition from state s_i to state s_j.
- An **observation cost** B_{ij} which is the likelihood of observing a candidate from state s_i.

For our task, the ball can be in one of two states $S = \{s_l, s_r\}$, either the left or right hand. The transition cost for a particular time t is computed as the L_2 distance between the left and right hand, $A_{ij}^{(t)} = ||p_l^{(t)} - p_r^{(t)}||^2$. The observational cost at time t is the L_2 distance between the current state and the proposed ball candidate as detected by YOLOv3, $B_{ij}^{(t)} = ||p_{ball}^{(t)} - s_{ij}^{(t)}||^2$. This is calculated for each ball candidate in the frame. The total cost for time t is then calculated as

$$C_{j,i}^{(t)} = \min_k C_{k,i-1}^{(t-1)} + A_{k,j} + B_{j,k}.$$

This cost is stored in the cost matrix T_{cost} and the state k associated with this cost is saved in the path matrix T_{path}.

The most likely path is produced following Algorithm 1. By optimizing over this cost function, any erroneous candidates produced by the object detector are ignored over time. This is visualized in Figure 2.6. In this frame, the ball is in the subject's left hand. The transition cost between the left hand state and the outlier candidate is much higher than the ball detected in the subject's hand.

During the task, the instructional video audibly counts the beats at which the subject must pass the ball (e.g. AND, one, AND, two, ...). As part of the rhythm score, the "AND" beat must coincide with the movement in between passes. This is most commonly performed by raising the ball slightly before dropping it into the other hand. We analyze the output of the Viterbi-based path to calculate a raw accuracy and rhythm score. For accuracy, the moments of ball transition are considered. A point for accuracy is given even if the pass is not in rhythm with the beat. Rhythm is based on two beats per ball

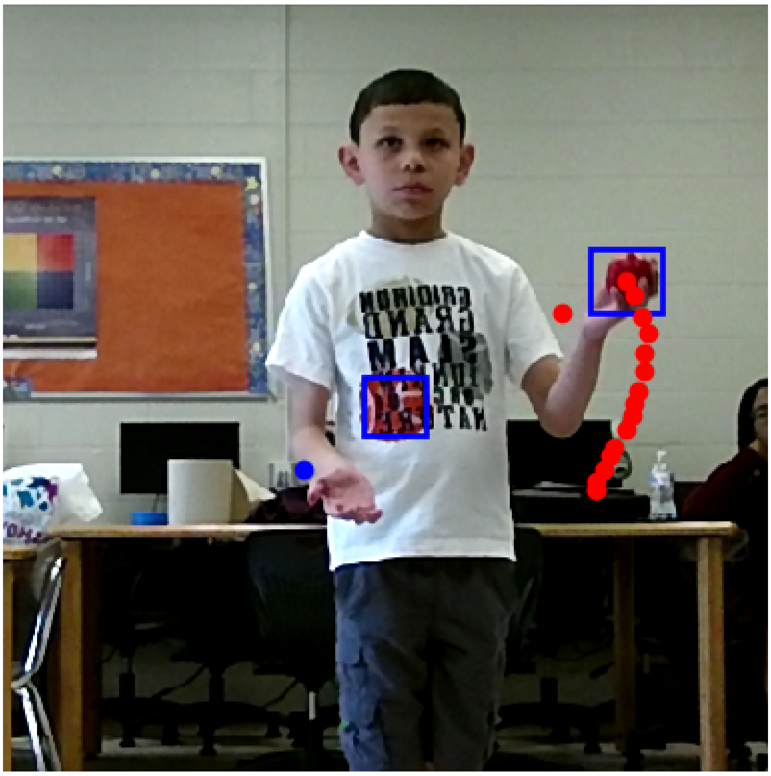


FIGURE 2.5: A frame with an additional detection. The red ball on the subject's shirt is considered, but rejected as the transition cost would be too high.

```
$z_T = \operatorname*{argmin}_k T_{cost}[k, T]$
$x_T = S_{z_T}$
for $i \leftarrow T, T-1, \dots, 2$ do
    $z_{i-1} \leftarrow T_{cost}[z_i, i]$ $x_{i-1} = s_{z_{i-1}}$
end
```

Algorithm 1: Calculating the most likely path.

pass. The first is when the hand carrying the ball is moved upward on the "AND" beat. The second is with the rhythm of the count.

To allow for some tolerance in what constitutes a beat that is on rhythm, our evaluator permits the movement window to be modified via a parameter β. The middle of the window is aligned with the time stamp at which the audible command is given. In practice, β is set to be the width of the window in number of image frames. Since the data is captured at 30 frames per second, a value of 1 is approximately 0.03 seconds. For the results reported in this dissertation, we selected a rhythm window of 30 frames unless stated otherwise.

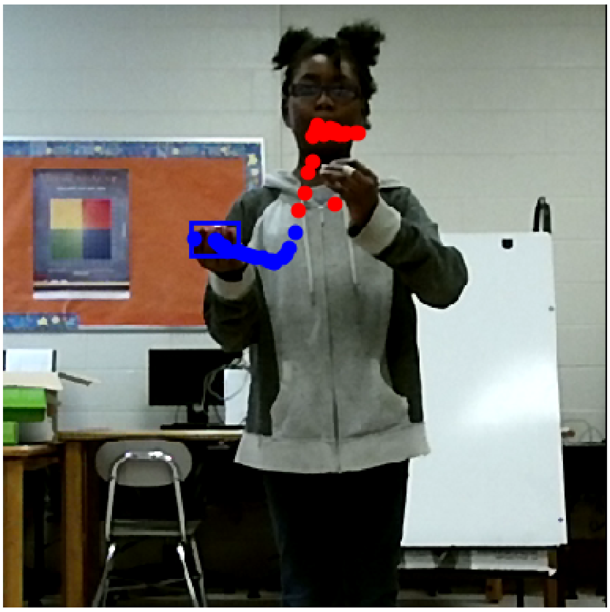


Figure 2.6: Most likely path visualized on the final frame of the subsequence.

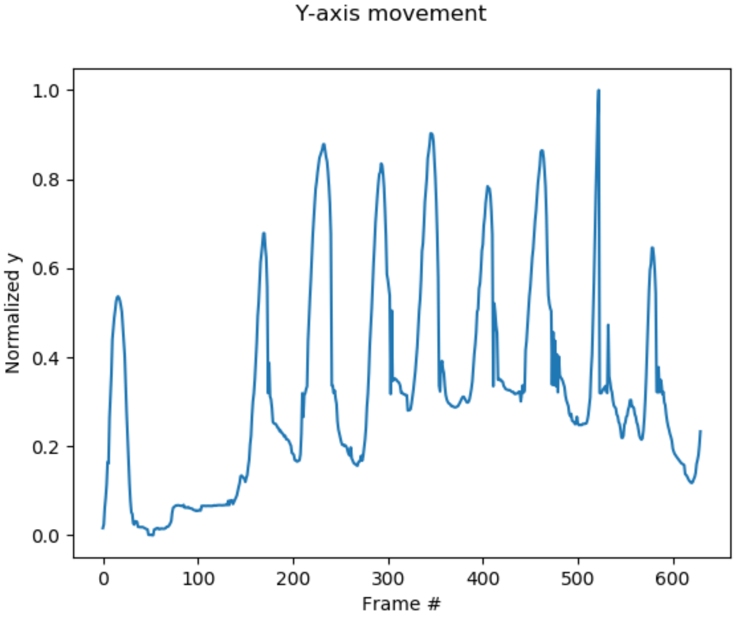


Figure 2.7: Normalized output trajectory of the ball as returned by the Viterbi-based method.

Figure 2.7 shows the ball path as returned by our method. The highest peaks in the graph coincide with the raising of the ball for the "AND" beats, and the valleys correspond to the passes. The exact timing of the instructions provided by the video are compared to this output in order to generate a raw score.

Subject	Predicted		Actual	
	Accuracy	Rhythm	Accuracy	Rhythm
1	8	9	8	14
2	8	3	8	14
3	8	9	8	14
4	8	9	8	14
5	6	5	8	16
6	8	15	8	10

Table 2.1: *Bag Pass* results of the Viterbi-based method.

Table 2.1 shows the results of our automated scoring approach as compared to the scores provided by expert reviewers. The passing accuracy is much higher than the rhythm using this method due to unexpected movements between each participant. If we relax the window parameter β, this method produces rhythm scores closer to that of the experts. For example, one subject would pass the ball from side to side instead of raising the ball first. Output of the detected "AND" and "PASS" frames are shown in Appendix A.

DeepGRU Model

The dynamic programming approach described in the previous session was our initial attempt at automatically evaluating this task. We also wanted to analyze the efficacy of modern deep learning methods for this task. Due to the relatively small dataset size, our model must be able to provide acceptable generalization performance from fewer samples. We also want to take advantage of useful data augmentation techniques for sequential data. Thus, we use the DeepGRU [38] model with augmented data synthesized using gesture path stochastic resampling [70].

Body pose keypoints are extracted from each frame using OpenPose [9]. These are the only features used to detect ball passes. A ball pass event occurs when the participant moves the hand holding the ball towards the other hand, makes the transfer, and moves back to the original position. In this case, the distance between the wrist points decreases until the transfer happens and increases again. Note that, using only the keypoints as input, the state of the ball is left unknown. However, this is of little consequence to the evaluation. What is most important is that the hand make the transition to the midpoint of the body and back. A hand raise even occurs when the participant moves the hand holding the ball towards the shoulder of the hand holding the ball and retreats back to the original position, where the distance between the wrist and the shoulder joint initially increases before decreasing as the hand returns back down. This movement produces a peak when plotting the y-value of the hand over time.

The model was trained and evaluated using sequence annotations for each subject and task. The sequences were labeled as one of the 3 classes. The model was trained using k-fold cross validation such that each fold is tested in a user independent manner. The overall segment accuracy is **61%**.

2.5 Sailor Step Task

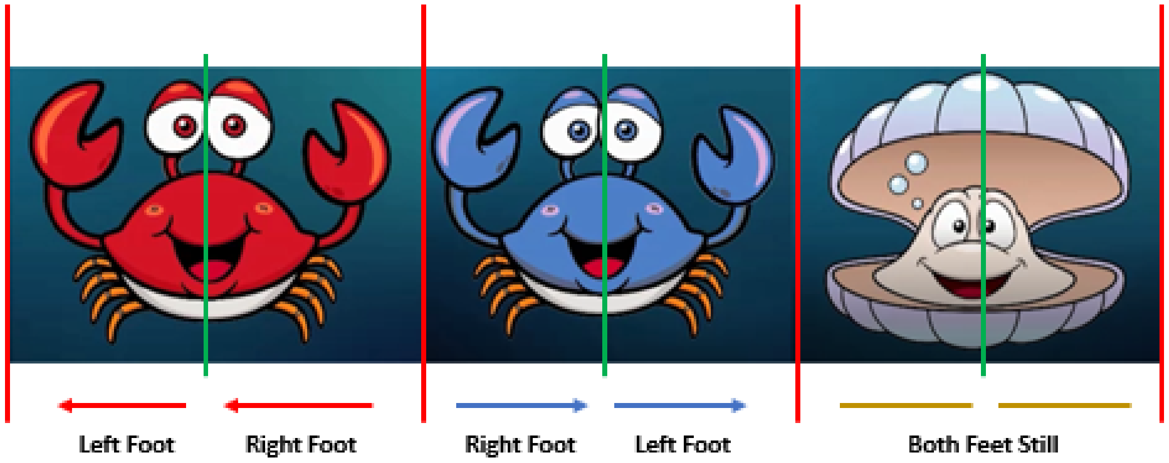


Figure 2.8: Sailor Step task rules. Children are instructed to perform a specific movement for each presented stimuli. Each segment has two beats; one for each foot movement.

The Sailor Step task is designed to assess visual cue processing while performing lower-body movements with accuracy, rhythm, and response inhibition. To make the task compelling and engaging to children, it is presented as a dance exercise, where the child must move according the instructions given in a presentation video. This video shows a predefined sequence of three different visual stimuli that appear on the screen for a given time: (a) a Red Crab, (b) a Blue Crab, and (c) a Happy Clam, as shown in Figure 2.8. Based on the rules, the child needs to move one step to the right when the Red Crab appears on the screen, one step to the left for the Blue Crab, and stand still if the Happy Clam appears. Each step is to be performed in two steps, corresponding to the rhythm of the song. This task requires the child to (a) remember the rules, (b) move accurately corresponding to each instruction, and (c) move in rhythm with the song.

The scoring approach considers three different scores: (a) visual accuracy, (b) visual response inhibition, and (c) visual rhythm. It consists of 21 total instructions which are presented during the song in a predefined and fixed order to ensure test-retest reliability. The visual accuracy score reflects the number of movements performed correctly. The raw score range for visual accuracy is $[0, 13]$. A point is awarded for a correct movement even if it is not performed in rhythm. The response inhibition score measures how the child responds during a Happy Clam instruction. This score is in the range $[0, 16]$. Partial points are also awarded to the participant if they are still for at least 1 beat in rhythm with the song. The rhythm score measures how many movements were performed to the rhythm of the beat. For instructions corresponding to left or right steps, a rhythm point is awarded for each step that is in rhythm. There are 26 possible raw rhythm points available corresponding to two beats per each of the 13 steps.

2.5.1 Experimental Approach

Given a video $\mathbf{X} \in \mathbb{R}^F$ of a subject performing the entire task, we segment $\mathbf{X}$ according to the timestamps of each prompt within the task. There are 29 prompts, 13 crabs and 16 happy clams, in total. All recordings can be segmented following the instructional prompts since they are given in the same order each time the task is performed. For the initial experiments, we annotated 15 recorded sessions, resulting in 334 annotated segments.

Segments for the slow version (60 BPM) of this task are 45 frames in length, which we use as the standard input length for our experiments. Segments for the faster version (100 BPM) do not meet the standard length and are padded with zero vectors for evaluation.

We find that the detected keypoints provided by the Microsoft Kinect V2 dataset are noisy, thus we use OpenPose to extract the keypoints from each frame [9]. OpenPose produces 25 keypoints for each human detected in each image. Since the task focuses on lower-body movement, we only use the keypoints from the hips down. In practice, we use 12 3D keypoints from each frame $x_i \in \mathbb{R}^{12\times 3}$.

Similar to the *Ball Drop* task, we evaluate the data using a PyTorch implementation of DeepGRU [38]. This model was chosen because of its ability to explicitly model temporal features via a recurrent design. Instead of using implicit visual features extracted from a CNN, we opted to use the body joint locations directly, as stated above. For training, we follow Maghoumi et al. [38] and use an implementation of gesture path stochastic resampling (GPSR) [70] to augment the data.

Considering the scoring guidelines and the different stimuli, the available annotations are [*LeftComplete*, *LeftIncomplete*, *RightComplete*, *RightIncomplete*, *Still*] given a "crab" segment and [*Still*, *HalfStill*, *NotStill*] given a "happy clam" segment. A label of *Other* was also used for random or extraneous movements for future analysis. Considering the task rules and the different stimuli (clams, crabs), our experimental approach includes the training of five different models which predict the following classes:

- $M : [LeftComplete, LeftIncomplete, RightComplete, RightIncomplete, Still, HalfStill, NotStill]$
- $M1 : [Left, Right, Still]$, where $Left = [LeftComplete, LeftIncomplete]$ and $Right = [RightComplete, RightIncomplete]$
- $M2 : [LeftComplete, LeftIncomplete]$
- $M3 : [RightComplete, RightIncomplete]$
- $M4 : [Still, HalfStill, NotStill]$

These models were selected considering the presented stimuli and the scoring guidelines for both accuracy and rhythm. One approach would be to use a single model M for all segments (stimuli-required movement). Another approach, as proposed in previous work [7], is a hierarchical one which considers the presented stimuli of the segment. Given a red/blue crab, the system uses model $M1$ to predict the direction of the movement $(Left, Right, Still)$. The predicted direction can be used to score accuracy, compared to the required direction (task rules). Given a predicted direction, in order to score for rhythm $(Complete = 2, Incomplete = 1, Still = 0)$, the system uses models $M2$ and $M3$ for left and right, respectively. Given a "happy clam" segment, the system uses model $M4$ which assigns an accuracy score $(Still = 2, HalfStill = 1, NotStill = 0)$.

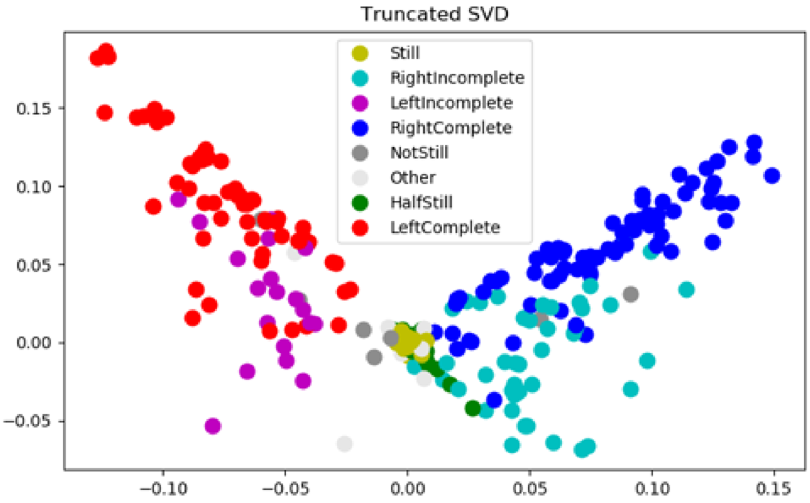


FIGURE 2.9: 2D projection of the data using truncated SVD

In order to get an insight of the class distributions and evaluate our classes selection, we applied truncated SVD for dimensionality reduction to visualize a 2D projection of the datapoints, as seen in Figure 2.9. We observe that there is a much clearer distinction between Right and Left (complete/incomplete) classes, compared to the Still classes $(Still, HalfStill, NotStill)$.

2.5.2 Results

We report the results of all model versions in Table 2.2. The individual confusion matrices for the combined models are shown in Figure 2.10. The confusion matrices for the standalone *Left, Right,* and *Still* models are shown in Figure 2.11. Across all models that consider *Still*, the accuracy is very high. This is expected and is intuitive when considering the trajectory of the detected body keypoints through time as the subject is still. None of the subjects evaluated completed failed a *Still* prompt. Likewise, there were no movements that were extraneous enough to label as *Other*.

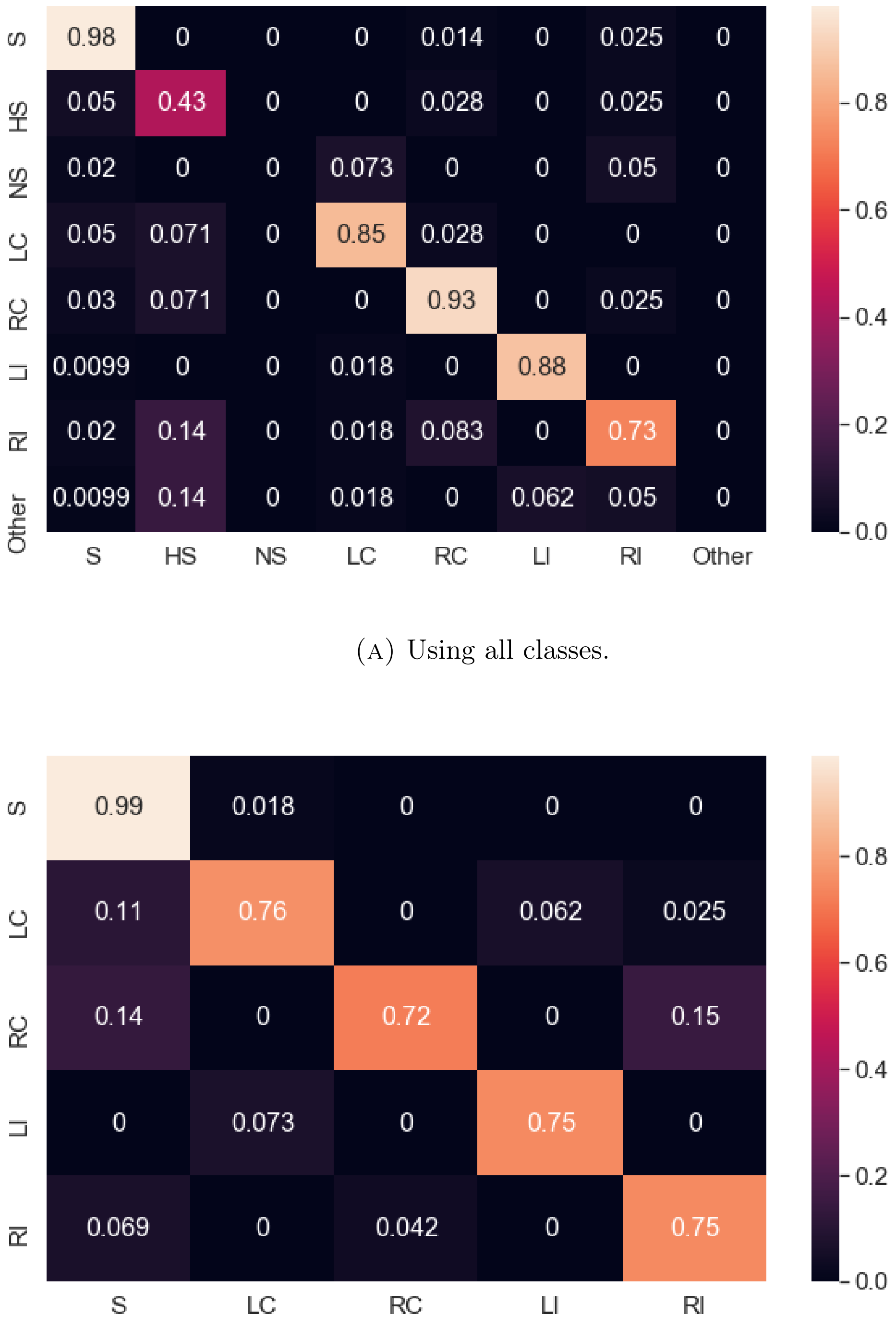
Model	Accuracy
M	83.7%
M1	83.1%
M2	88.7%
M3	92.9%
M4	86.2%

Table 2.2: Summary of classification accuracy of all model types.

2.6 Finger Tap Task

The *Finger Tap* task of ATEC is one of six physical exercises designed to measure rapid and sequential movements. Contact between the index and thumb are considered for tapping. Subject performance is evaluated based on rapidity and fluidity. Rapidity is determined by the total number of correct finger taps within 10 seconds. The fluidity score measures the smoothness and accuracy of motion. A score of 0 is given if the subject stops and does not restart the movement or switches to another movement and does not change back to the correct movement. A single point is given if the participant stops the motion but continues finger tapping. Two points are given for 10 seconds of continuous correct movement. In this section, we present a baseline assessment and evaluation of the *Finger Tap* task. The goal of this preliminary work is to analyze the data recorded from eligible participants, formulate a method of automatic scoring, and evaluate the effectiveness of current computer vision and machine learning methods related to scoring this task.

We formulate the automatic scoring of this task as a binary classification problem, where the model predicts whether or not the fingers are open or closed. An example of



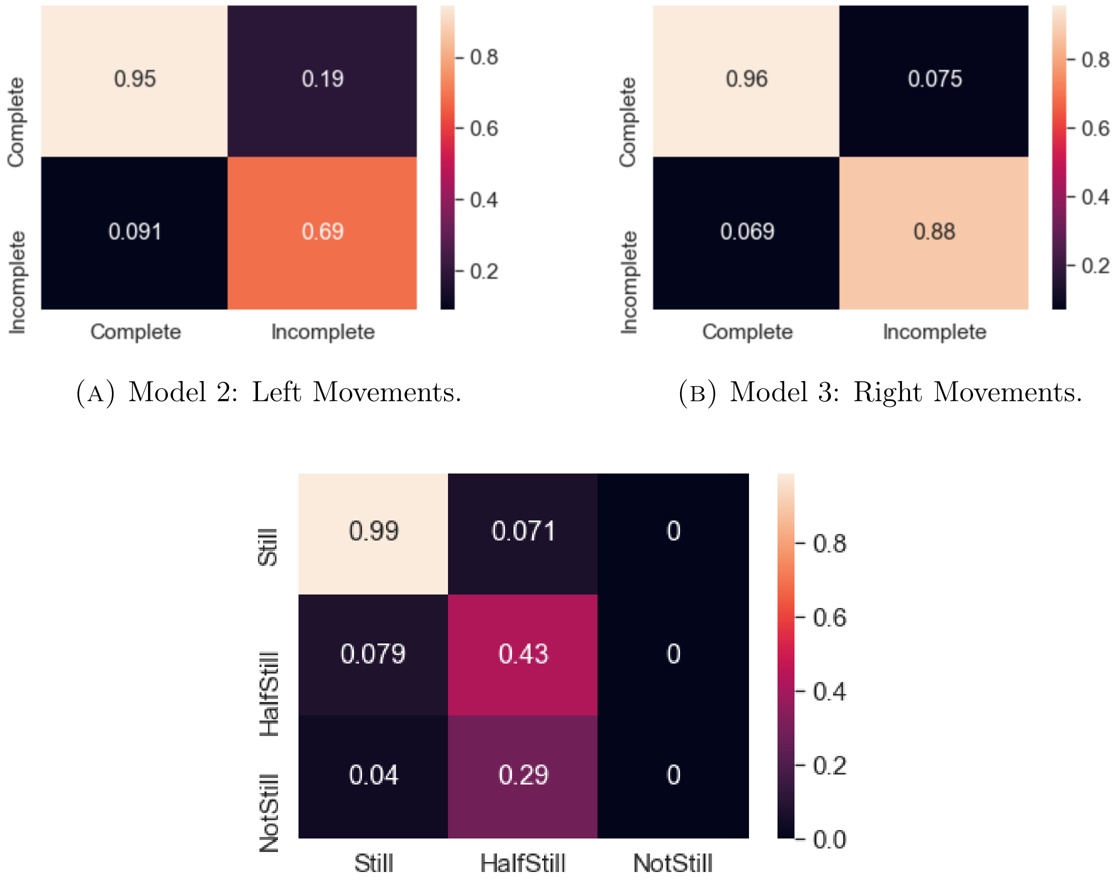
	S	HS	NS	LC	RC	LI	RI	Other
S	0.98	0	0	0	0.014	0	0.025	0
HS	0.05	0.43	0	0	0.028	0	0.025	0
NS	0.02	0	0	0.073	0	0	0.05	0
LC	0.05	0.071	0	0.85	0.028	0	0	0
RC	0.03	0.071	0	0	0.93	0	0.025	0
LI	0.0099	0	0	0.018	0	0.88	0	0
RI	0.02	0.14	0	0.018	0.083	0	0.73	0
Other	0.0099	0.14	0	0.018	0	0.062	0.05	0

(A) Using all classes.

	S	LC	RC	LI	RI
S	0.99	0.018	0	0	0
LC	0.11	0.76	0	0.062	0.025
RC	0.14	0	0.72	0	0.15
LI	0	0.073	0	0.75	0
RI	0.069	0	0.042	0	0.75

(B) Model 1: Does not consider *HalfStill* or *NotStill*.

Figure 2.10: Confusion matrices for full models.



(A) Model 2: Left Movements.

(B) Model 3: Right Movements.

(C) Model 4: Still Movements.

FIGURE 2.11: Confusion matrices for each split model evaluated.

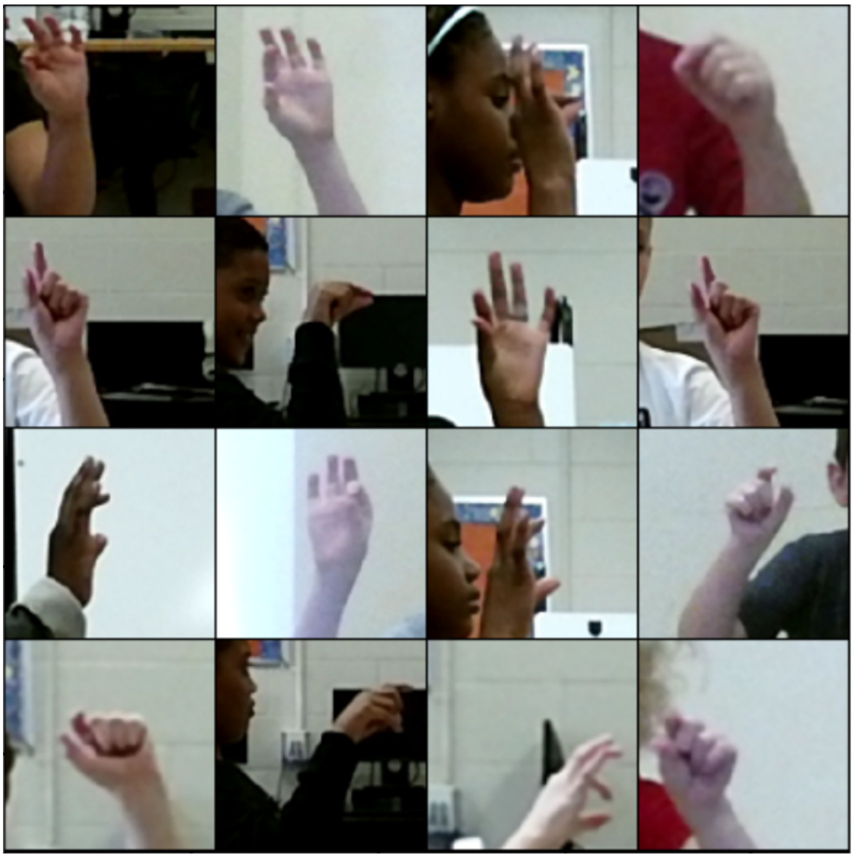


FIGURE 2.12: Hand images cropped from subjects performing the *Finger Tap* task.

each case is shown in Figure 2.13. Training is performed using individual frames recorded from both the left-hand and right-hand version of this task. Each recorded session consists of approximately 400 frames. Each subject performs this task for both the left and right hands recorded by 2 cameras. Additionally, the subjects return at a later date for a followup evaluation. Thus, the total number of recorded frames for each subject is about 1600.

For the purpose of automatic scoring, a machine learning model would only need to consider a cropped image of the hand. Even with an image resolution of 1920 × 1080, this presents a challenge based on the captured data. Due to the distance between the camera and subjects, each cropped hand has a resolution of less than 100 × 100. An initial consideration was the use of hand pose estimation models to predict the keypoints of the thumb and index fingers. A simple distance check between the two keypoints would determine successful taps. However, the resolution of the input hand images is too small,

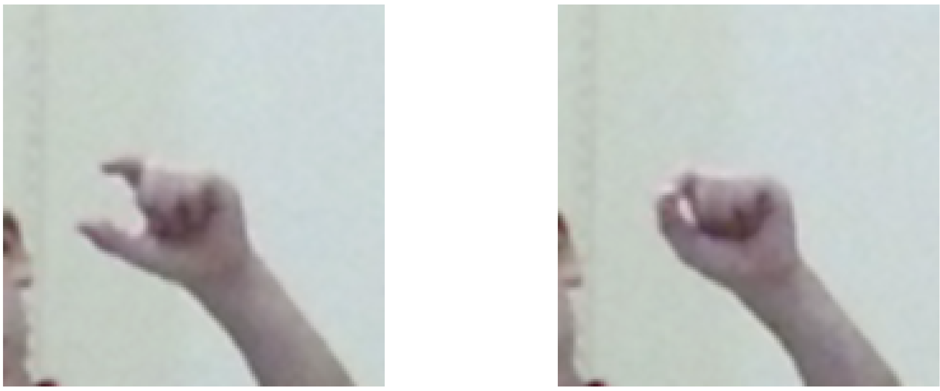


FIGURE 2.13: Open (left) and close (right) classes for *Finger Tap*.

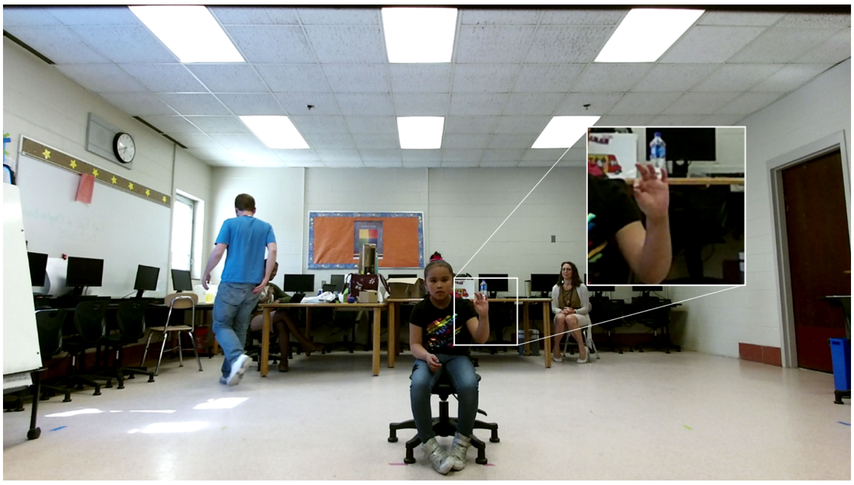


FIGURE 2.14: Visualization of hand detection and cropping.

resulting in noisy outputs. Figure 2.12 provides some samples of the hands cropped from participants.

2.6.1 Baseline Model

Given an image $X \in \mathbb{R}^{h \times w \times 3}$, a simple hand detector is used to determine the centroid, or palm, of the hand. A small bounding box centered on the predicted location is used to crop the hand from the original image. The size of the bounding box is chosen to allow data augmentation through scaling, rotation, and translation. A visual example of data pre-processing is shown in Figure 2.14.

Our image classification network is based on VGG16 [62], a popular image-based deep learning model. The model is first pre-trained on the ImageNet dataset, a large-scale image classification dataset with over 1.5 million samples of 1000 unique classes [16].

Predicted Taps	Actual Taps	Frame Accuracy
909	922	85.75%

Table 2.3: Cumulative frame prediction results for 12 subjects and 25 recordings.

After pre-training, the final layer is discarded since our task only considers 2 classes. The network is fine-tuned using the cropped hand images from our dataset.

To count the number of predicted taps in each sequence of frame predictions, a sliding window approach is used. The window begins when an open hand is detected. Once a closed hand is detected, the model waits for the open frame again before classifying the subsequence as a tap.

2.6.2 Results

Without any additional filtering or smoothing, our model automatically predicts the state of the hand in each frame of an input video. We report the results from 12 subjects using k-fold validation in Table 2.3. The predicted taps are the taps that the model estimated based on analyzing the sequence of predictions corresponding to the input video. The actual taps are the manually annotated number of taps. Raw frame accuracy is the percentage of correctly predicted frames compared to the frame-level manual annotations. The cumulative results are listed at the end of the table.

Chapter 3

Self-Supervised Learning for Hand Pose Estimation

Deep learning based 3D hand pose estimation requires large amounts of data for training. Fully supervised methods provide reasonable accuracy under specific conditions but require 3D annotations for each individual frame in the dataset. Providing such annotations is an expensive task that may be infeasible for many novel applications. This chapter discusses self-supervised methods for training 3D hand pose estimation models with little or no joint annotations.

The self-supervised component is based on a 3D hand model and will generate a point cloud representation of the predicted pose. Parameter choices will be evaluated to determine the best representation for hand pose and shape. The predicted hand shape will be compared with the input depth image as a means of supervision which can be used in parallel with joint annotations. An initial version of this framework is reviewed on a standard hand pose estimation dataset.

In this chapter, self-supervised training is defined in the context of hand pose estimation. A self-supervised framework for learning from unlabeled data is evaluated. By estimating the interpretable parameters of a hand model, a computer graphics-based generator creates samples of the pose and shape of the hand. This formulation only requires that the choice of estimator output be hand model parameters that can be interpreted by the generator. This allows new and effective estimator architectures to be explored while maintaining the property that it works with the chosen form of self-supervision.

Each method is benchmarked on synthetic samples as well as a standard 3D hand pose estimation dataset.

3.1 Hand Pose Estimation

Hand Pose Estimation has and continues to be an active research area. Recent methods have primarily been deep learning based. The work in this dissertation follows these approaches. For a review of earlier works and alternative methods, we refer the reader to [23]. The availability and affordability of depth sensors has made 3D keypoint estimation the primary task. A common input modality of hand pose estimation methods is depth as it represents the required 3D information. Although RGB input is used in some methods [52, 86, 8, 20, 45], most recent approaches work with depth data [67, 68, 49, 30, 17].

Yuan et al. present a comprehensive review of depth-based hand pose estimation [83]. The challenges and limitations they report align with the goals set forth in this thesis. First, they report that current methods are still unable to maintain high accuracies for extreme viewpoints. This is partly because of a lack of annotated data for such scenarios. A model that lacks visual features of these scenarios would likely fail as well. Second, they note that discriminative methods do not generalize well to unseen hand shapes. Finally, they observe that methods which explicitly encode structure result in significantly reduced errors on both visible and occluded joints.

One of the first deep learning-based approaches for depth based hand pose estimation was reported in [73]. They used a CNN to extract 2D heatmaps of the joint locations instead of directly regressing the keypoints. The simultaneous release of the NYU Hand Pose dataset kicked off a significant increase in hand pose estimation methods. Citing the success of human body pose estimation methods, Oberweger et al. propose a CNN-based model that directly regresses joint positions [50]. Their accuracy is improved by predicting the parameters of a low dimensional space before mapping back to 3D joint locations. They additionally include a refinement network which improves upon the coarse predictions of the initial network. Ge et al. note the limitation of using 2D heatmap predictions for 3D keypoint estimation [27]. Their proposed method uses multiple CNNs in parallel to learn from multiple viewpoints of the same hand. In this way, the output 2D heatmaps can be fused in such a way that 3D information is extracted. Noting that models for hand pose estimation were quickly becoming complex, Guo et al. introduce a single simplified model trained end-to-end. Their Region Ensemble Network segments the input image into separate regions with a fully connected layer for each region.

3.2 Learning from Unlabeled Data Using Self-Supervision

A common observation of hand pose estimation research is the difficulty and expense of providing manual annotations for novel datasets. Methods attempting to overcome this challenge either turn to synthetic data or self-supervision. Few methods have proposed a form of self-supervision for hand pose estimation, but progress has certainly been made in recent years. We review current attempts that utilize some form of self-supervision and note that no approaches have been able to achieve performance competitive with fully supervised methods without at least some joint annotations.

Wan et al. assume a one-to-one mapping between image space and pose space [77]. The pose space is modeled using a Variational Auto-Encoder while the image space is trained using Generative Adversarial Networks. The gap between the latent pose and image space is bridged by learning an additional mapping function. Their GAN formulation allows additional self-supervision by combining unlabeled and labeled images. This effectively bridges the domain gap between real and synthetic examples. It does not allow for completely self-supervised training.

Instead of learning an additional mapping, Spurr et al. learn a latent space that combines both pose and image space by training two encoder-decoder pairs [64]. Each encoder network transforms the input into a single latent representation that can be converted to either modality using the corresponding decoder. They argue that not all joints are articulated independently, suggesting that dimensionality reduction is effective.

Although their method allows for semi-supervised training using unlabeled RGB images, it cannot predict pose without at least training the pose decoder.

Poier et al. observe that the pose of the hand as seen in one view is predictive for the appearance of the hand in any other view [55]. This observation unlocks a virtually infinite number of training examples by using multiple views of the same hand. In practice, datasets collected using a stereo camera setup provide multiple views of the same hand which act as the input and target. A pose estimation network maps the depth image of one view to a set of pose parameters. These parameters are used as input to an image and pose decoder. The image decoder produces a rendering of the hand as seen from a different view. The rendered and original depth images are compared using an L1 reconstruction loss. Simultaneously, a pose decoder is trained using the 3D joint annotations. This formulation allows semi-supervised learning with unlabeled samples collected in a stereo setup. Similar to [64], without training the pose decoder with joint annotations, the 3D keypoints cannot be estimated.

Although attempts using synthetically generated images are useful for augmenting the dataset and self-supervised learning, they do not accurately model the features seen in real data. In subsequent work, Poier et al. propose a method for bridging the real-to-synthetic domain gap [56]. To address this gap, they learn a mapping from real features to synthetic features using a discriminator function.

Cai et al. propose a semi-supervised method using a depth rendering network [8]. Instead of using a 3D hand model, a depth image is synthesized through several transposed convolutional layers. Even in a weakly supervised setting, their method still requires supervision in the form of 2D heatmaps.

Abdi et al. advance on the idea of a shared latent space to resolve the real-synthetic domain gap [1]. Their model can be trained with real, synthetic, and unlabeled data. Similar to other methods that learn a shared latent space [64, 77, 55], the model requires at least some training with joint annotations.

Recently, Li et al. adapt the self-organizing network in [37] for point cloud representations of the hand. The use of a parameterized hand feature decoder recovers point cloud representations of the input feature vector. The generated point cloud can be compared with the input to learn in a self-supervised manner [10].

The self-supervision scheme suggested in this dissertation is inspired by Dibra et al. [19, 20]. In their initial work, they use an estimator network based on AlexNet to produce the pose parameters of a 3D hand model. The vertices of the posed hand mesh are first sampled before being rendered as a depth image. There is an observed discrepancy between the generated and real images, so a point cloud is sampled from the target depth image before being rendered using the same function. The generated and target depth images are compared using L1 loss. They note that their model could adapt to individual hand shapes but do not perform any experiments related to that task.

3.3 Learning from a Cold Start

The cold start problem is loosely defined as modeling the distribution of hand poses from unseen and unlabeled data. Reaching human level accuracy from a cold start obviates the need for joint annotations. In this section we review hand pose estimation methods which attempt to tackle this problem. We note that so far no hand pose estimation methods address the cold start problem, in which the model is trained from scratch without joint annotations.

Some methods propose a framework which can serve as the basis of a cold start solution. Dibra et al. approach this by using a differentiable rendering pipeline with a 3D hand model [19, 20]. For training real data, they use a self-supervised loss consisting of a depth component. An additional physical and collision loss are included to encourage plausible configurations of the hand. After pre-training on a synthetic dataset, they train on real data using only unlabeled samples.

The state-of-the-art approach which addresses the cold start problem is that of Wan et al. [78]. They use a hand model approximated by spheres for model fitting. A synthetic depth image is rendered using a differentiable rasterizer. The rendered depth is compared to the unlabeled depth image in lieu of explicit joint annotations. Their model is initially

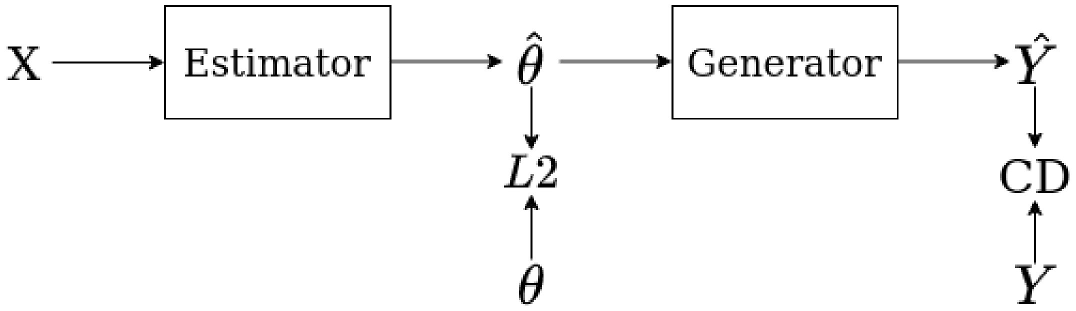


Figure 3.1: Abstract layout of the framework used. The estimated parameters $\hat{\theta}$ can be optionally compared if the ground truth is available. The target image Y is compared to the generated image using a loss such as Chamfer Distance.

trained on synthetic data with joint annotations before being trained on unlabeled real samples from the NYU Hand Pose Dataset [73]. Although their method does not surpass other self-supervised methods, they are currently the only work to approach competitive accuracy on a standard dataset without the use of joint annotations.

3.4 Method

We propose a general framework for hand pose estimation trained with self-supervised learning. The entire method can be viewed as an autoencoder in which the decoder is a series of modeling functions used in Computer Graphics. Note that, under this general formulation, it is possible to use any form of generator, such as Generative Adversarial Networks (GAN) [28]. The only requirement for the encoder is that it accepts some visual representation of the hand as input. This representation can be RGB, Depth, or Point Cloud. Instead of directly estimating the 3D keypoints of the hand, the encoder produces hand parameters used in articulating and shaping a 3D hand model. The choice of parameters can vary depending on the desired output. For example, if an estimate of the shape of the hand is required, then the encoder may also produce shape and scale parameters.

3.4.1 Cold Start

With no prior information about plausible hand poses and shapes, a randomly initialized estimator has little chance to converge. Prior knowledge of the hand can be encoded in several ways. One simple way of doing so is to add a loss which penalizes the joint angle predictions if they are not within a set of acceptable physical boundaries. For model-based methods, the joint angle loss is explicit. Parameterized methods typically account for physical boundaries using a prior term, such as the loss term used by Wan et al. [78].

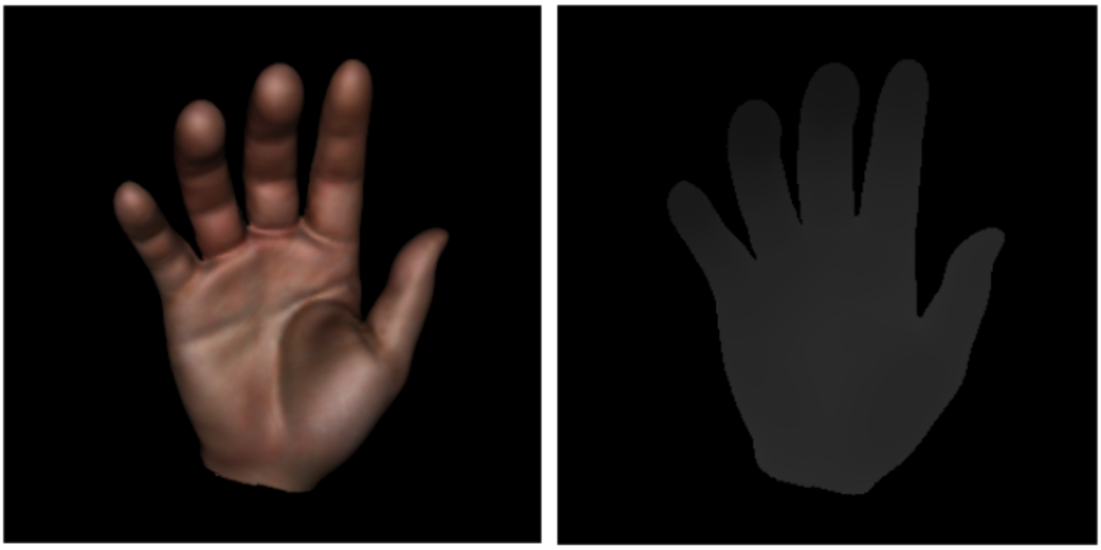
Another source of prior knowledge is the generator itself. The shape and pose of the hand model can be randomly modified to produce plausible synthetic depth or point cloud samples. Since the kinematic structure of the hand is included in the generator, the keypoints are automatically generated.

Similar to [56], we take advantage of the fact that the pose is consistent between two views of the same hand. We enforce our model's predictions from each view to be consistent except for the parameter representing global orientation, as this will naturally differ between two separate perspectives.

To aid model convergence in the beginning of training, we compare each sample image with a representation of our hand model in its default pose (seen in Figure **??**). If the default estimate provides a lower loss, that error is used to update the weights during backpropagation.

3.4.2 Hand Model

The hand model used in the following experiments is based on libhand [61] consisting of roughly 32,000 vertices. It is modified slightly to match the keypoints used by several standard datasets including BigHand2.2M [82] and RHD [86]. The wrist has also been removed so that the only points produced belong directly to the hand. The model is defined by a skeleton, a mesh, and a list of vertex-bone weights. The skeleton defines the hierarchy of joints along with the position and orientation of each joint relative to their respective parent joints. For the mesh, the position, normal vector, and texture



(A) RGB output of hand model. (B) Depth output of hand model.

FIGURE 3.2: Sample output of the 3D hand model used in this work.

coordinates are given for each vertex. Finally, the mesh definition also includes a list of vertex-bone weights which define the correspondence between all the vertices and each joint.

3.4.3 Estimator

The only requirement for the estimator is that it be a learnable, deep neural network that outputs parameters used by the hand model and generator. For these experiments, the joint angles are represented as Quaternions. Any other representation for angles, such as Euler angles, would be sufficient as well. ResNet-18 [31] is chosen as the feature extractor and estimator. Instead of producing a final vector of classification probabilities, the final layer of ResNet is modified to output $N_j \times F$ values, representing a Quaternion offset along with other parameters such as position and scale. The exact choice of features is experimental and two iterations are evaluated in this work: one with orientation and position offsets and another producing orientation offsets and a scale parameter for each joint. The ResNet network is trained completely from scratch.

We also use an estimator that works directly with point cloud data extracted from depth images. Besides producing joint angle and shape parameters, this estimator also predicts part segmentation for each point in the input. The architecture follows Qi et al. [57]. Given point cloud input $X \in \mathbb{R}^{N \times 3}$, the point set feature network abstracts the points in a hierarchical manner. This produces a reduced set of 3D points as well as a set of corresponding features. The segmentation layers use the 3D points and features corresponding to the set abstraction layers from the feature learning stage. For estimating hand model parameters, the 3D points and features from the final feature extraction network are concatenated and used as input to a linear network which predicts the joint and shape offsets for each deformable joint in the hand model.

3.4.4 Generator

The generator produces 3D keypoints and a point cloud representation of the visible shape by following a standard rendering process. The 3D keypoints are calculated from the hand model using forward kinematics. The point cloud is generated by first transforming the mesh using the estimator parameters and then applying z-culling. First, the offset Quaternions are combined with the default joint orientations using the Hamilton product

$$\begin{aligned}
\mathbf{q} * \mathbf{v} &= q_1 v_1 - q_2 v_2 - q_3 v_3 - q_4 v_4 \\
&= (q_1 v_2 + q_2 v_1 + q_3 v_4 - q_4 v_3) i \\
&= (q_1 v_3 - q_2 v_4 + q_3 v_1 + q_4 v_2) j \\
&= (q_1 v_4 + q_2 v_3 - q_3 v_2 + q_4 v_1) k.
\end{aligned}$$

This results in a transformation from local bone space to object space. Since the mesh is given in the default pose in object space, inverse transformations $\mathbf{D}^{-1}$ are applied for each joint which transform the mesh to local bone space. The output orientations are converted to 4x4 matrices M_i for each joint so that an affine transformation of the hand mesh can be applied. Using Linear Blend Skinning (LBS), each vertex v_j of the mesh is transformed as follows:

$$v_j^{'} = \sum_i w_{ij} M_i D_i^{-1} v_j.$$

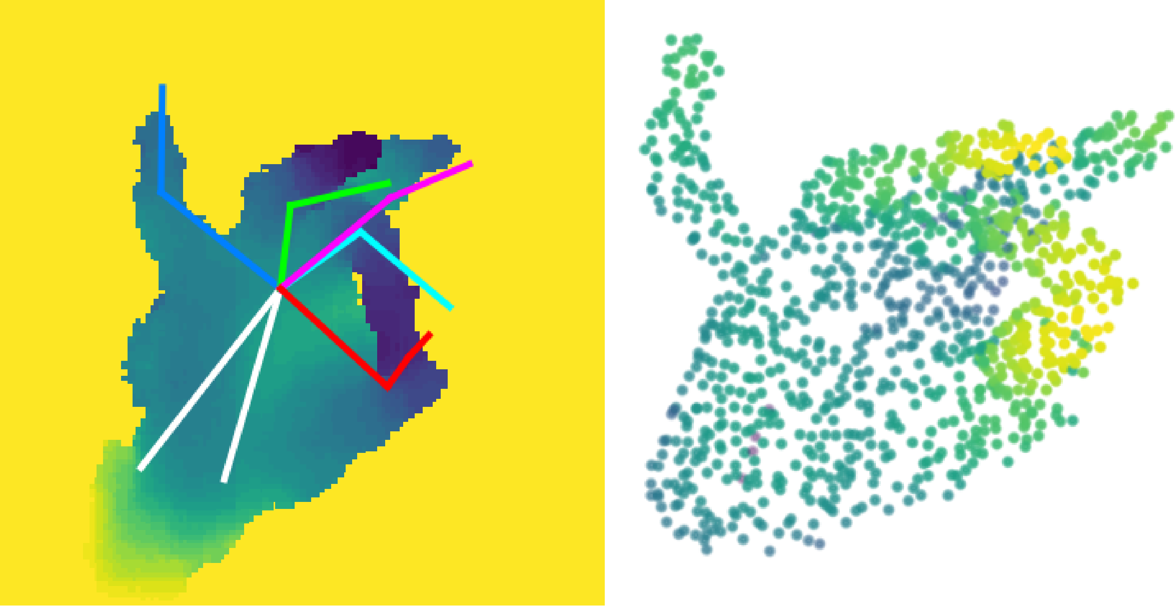


FIGURE 3.3: Sample from the NYU Hand Pose dataset with coordinate predictions (left) next to the generated point cloud from our model (right).

At this point the computed vertex mesh includes all 3D points of the hand. This does not resemble the target point cloud because no camera transformations or culling have been applied. To reduce the complexity of the experiments with the synthetic dataset, the known model-view matrix is applied to the generated point cloud such that the hand is in the same relative position as the target data. The visible points of the mesh are calculated using a differentiable depth culling function. Since the hand model mesh defines a normal vector for each vertex, the transformation matrices can be applied to these vectors before filtering those that face away from the virtual camera. This function produces a point mask that is applied to the full mesh. Using a mask ensures that only the visible points are used during the gradient calculation of the backward pass. Figure 3.3 shows an input sample from the NYU dataset [73], the model's keypoint prediction in 3D space, and the generated 3D point cloud.

3.5 Experiments

The depth-based model is evluated in two different scenarios, described in the next subsection, on the NYU Hand Pose dataset [73]. A common evaluation protocol for self-supervised methods is adopted in which the model is trained using a varying amount of labeled data. This evaluation serves as an indicator of performance of the self-supervised component. Aside from establishing an expectation of performance, this benchmark provides a clear path to understanding the individual sub-problems and assumptions made by comparing how they affect training.

3.5.1 Training

Two different training procedures are used to study the effectiveness of the full model. The first is the fully supervised setting in which only joint annotations are used with MSE loss. The second training procedure limits the amount of labeled data and combines a point cloud loss using Chamfer distance. Specifically, the first model is trained using only 75% of the labeled joint annotations, the second using only 50%, and the final model receiving only 25%.

Self Comparison

Two different versions of our model are trained. The first follows [39] and uses a scale parameter for the bone lengths of each joint. The second uses a position offset. This is similar to how objects would be defined in a Computer Graphics application. The position offset parameters are passed through a scaled hyperbolic tangent function:

$$y = \tanh(x) * \alpha$$

such that the value for each offset dimension is constrained within the range $[-\alpha, \alpha]$ in local space.

Method	**25%**	**50%**	**75%**	**100%**
Wan et al. [77] (No Self-Supervision)	16.5	16.2	16.3	15.8
Wan et al. [77]	16.1	16	15.9	15.5
Abdi et al. [1] (No Self-Supervision)	-	16.5	-	16.3
Abdi et al. [1]	-	16	-	15.83
Chen et al. [10]	**14.9**	**14.1**	**12.8**	11.2
Poier et al. [56]	-	-	-	**9.5**
Ours (No Self-Supervision)	18.92	18.03	15.67	14.21
Ours (Self-Supervision w / Bone Length)	17.31	16.09	14.66	14.74
Ours (Self-Supervision w / Position Offset)	16.96	15.71	14.59	14.05

Table 3.1: Mean Joint Error (mm) of the NYU Hand Pose test set.

Using a position offset gives the model more flexibility to match the shape of the hand defined by the annotations. As a result, this iteration of our model performs the best among our self-comparisons. With access to all labeled training samples, the position offset model performs better than the fully supervised component. This could be attributed to increased generalization using the hand shape data extracted from the point cloud. However, more experiments must be conducted to determine if these results are statistically significant.

As seen in Table 3.1, the self-supervised component of each model iteration is more stable to a decreasing number of labeled samples as compared to a fully supervised iteration. The relative increase in error from using 100% to 25% labeled samples is 33% for the fully supervised bone length model, where as adding the self-supervised component reduces this increase in error to 17%.

3.5.2 NYU Benchmark

The NYU Hand Pose dataset consists of 72,756 RGB, depth, and synthetic images captured from 3 different viewpoints for a total of 218,268 unique frames. It is standard within the hand pose estimation community to only train using the front camera. Semi-supervised approaches that use multiple viewpoints will necessarily use a different protocol.

Data augmentation is not always performed for this benchmark and we report results using no additional augmentation. Following [49], we pre-process the input by bounding the hand in a cube centered on the hand's center-of-mass. The depth is normalized between $[-1, 1]$ and background values are set to 1. The 3D keypoints are normalized based on the bone length of the first two joints of the index finger. This normalization value is saved for each sample for fair comparisons in the original 3D space.

We compare our results with several state-of-the-art methods which use some form of self-supervision component. Currently the best performing method is by Poier et al. [56]. They do not report exact accuracies at 25%, 50%, and 75% as reported in other works. Instead, they measure by consecutive powers of 10. Using 10,000 labeled samples, which is roughly 13.7% of the data, they report 9.9 mm accuracy. They use the labels from the front camera only, but use the provided synthetic depth images from all 3 viewpoints in their self-supervised component. They additionally perform augmentation on all input samples by randomly rotating, adding white noise, and adding a position offset.

Matching Evaluation Joints

Since our hand model is defined by 16 joints, the keypoints calculated using forward kinematics are different from the 14 evaluation joints defined in [73]. This discrepancy is observed in other works that use a predefined hand model. Malik et al. report that they select 16 closely matching points from 36 available NYU annotations for training [40]. Dibra et al. estimate the minimum error for each joint between their predictions and the ground truth [19]. This is calculated over the entire training set. Eleven of the NYU joints can be closely matched with joints in our hand model. The 2 wrist keypoints and palm keypoint must still be accounted for. Instead of estimating 3 additional points, we select them from the original mesh and anchor them to the root joint's transformations. Thus, the original points can still adapt to pose and hand shape while providing a fair comparison on datasets that do not use the same keypoints as our model.

This discrepancy between the keypoints given raises an interesting question about what level of accuracy is acceptable for each keypoint, especially when generalizing to new data. A reasonable standard would be to only select keypoints that have the lowest variation.

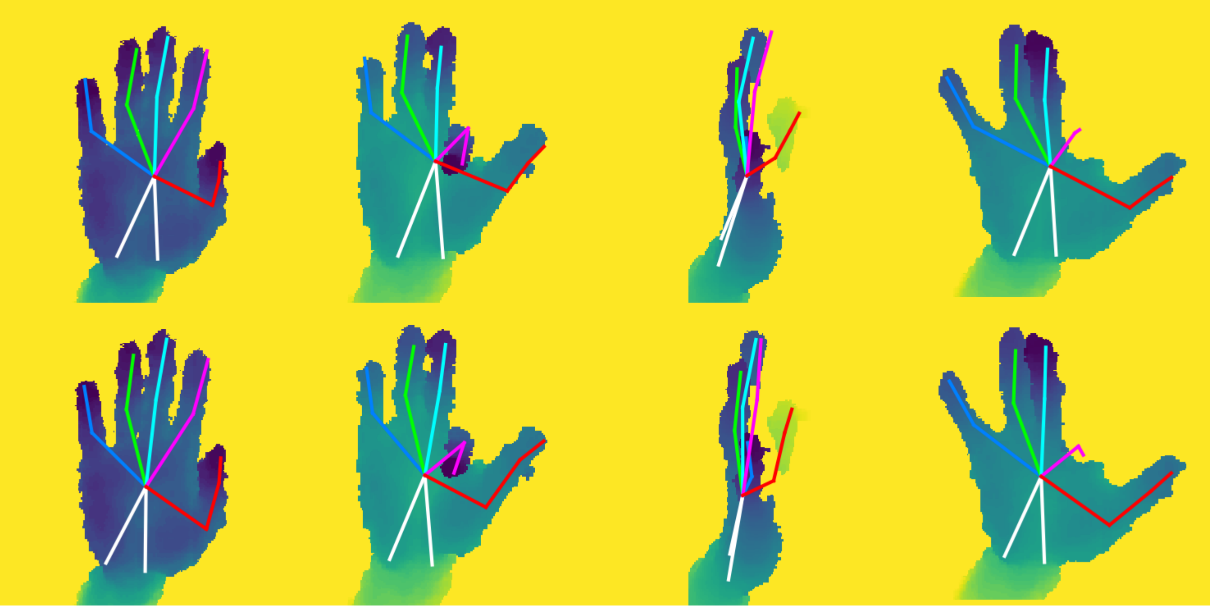


Figure 3.4: Examples of successful predictions (top) of our fully supervised model with shape self-supervision versus the ground truth (bottom).

For example, the tip of an adult human finger may have a width of 20 mm and length of 30 mm. At this size, a detected keypoint could be $\lesssim$ 18 mm while still being conceptually correct. Selecting the fold between joints as the keypoint would reduce the variation in length to a negligible amount, resulting in an approximate maximum acceptable error of $\lesssim$ 4 mm.

Qualitative Results

We review the predictions as compared to the target annotations to identify where our model is performing well versus where it fails. Figure 3.4 shows examples of where our fully supervised model with shape self-supervision performed well. The model predicted reasonably well even in cases of missing data, as seen in the two rightmost examples.

Figure 3.5 shows several failure cases of our fully-supervised model with shape self-supervision. From left to right, the first example exhibits several self-occlusions of the ring and pinky fingers as well as the wrist points. The prediction shows some awareness of the hand's whole orientation, but fails to make a reasonable prediction of the fingers. The second example shows a case where the index finger is isolated in clear view, but the model fails to make a good prediction. The rest of the finger predictions are bent

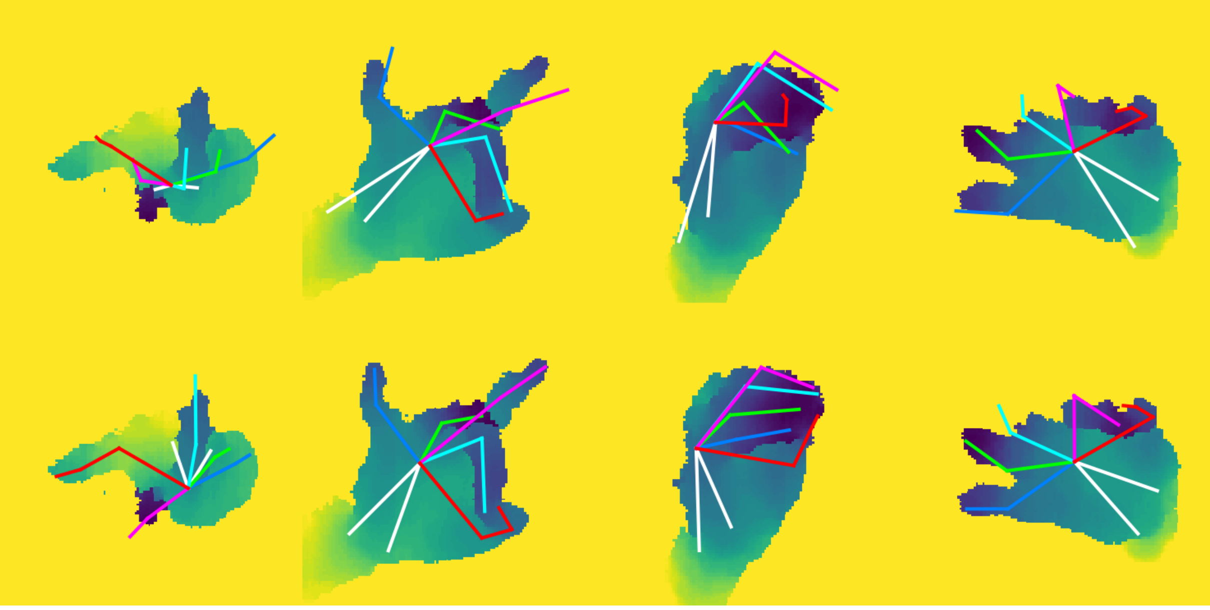


Figure 3.5: Examples of unsuccessful predictions (top) of our fully supervised model with shape self-supervision versus the ground truth (bottom).

forward as seen in the sample, but the entire hand prediction is offset. In the rightmost example, our model makes a reasonable prediction for the middle finger for which the data is missing. However, the prediction for the index finger does not reach the tip.

We also look at how the model's predictions degrade as more joint annotations are removed. Figure 3.6 shows predictions from 4 samples on the NYU test set. The top samples are predictions made when 100% of the joint annotations are available. The second, third, and fourth row depict predictions for 75%, 50%, and 25% available joint annotations, respectively.

Cold Start

When no annotations are made available, the model must be trained solely in a self-supervised manner. We train our model using synthetic, multi-view data provided as part of the NYU dataset. We enforce multi-view consistency by calculating the $L2$ loss between the estimated parameters of both input views. Additionally, a generated sample of the hand in a default pose is compared with each input view. We found that the addition of the default comparison was helpful in introducing prior knowledge at the early stages of training.

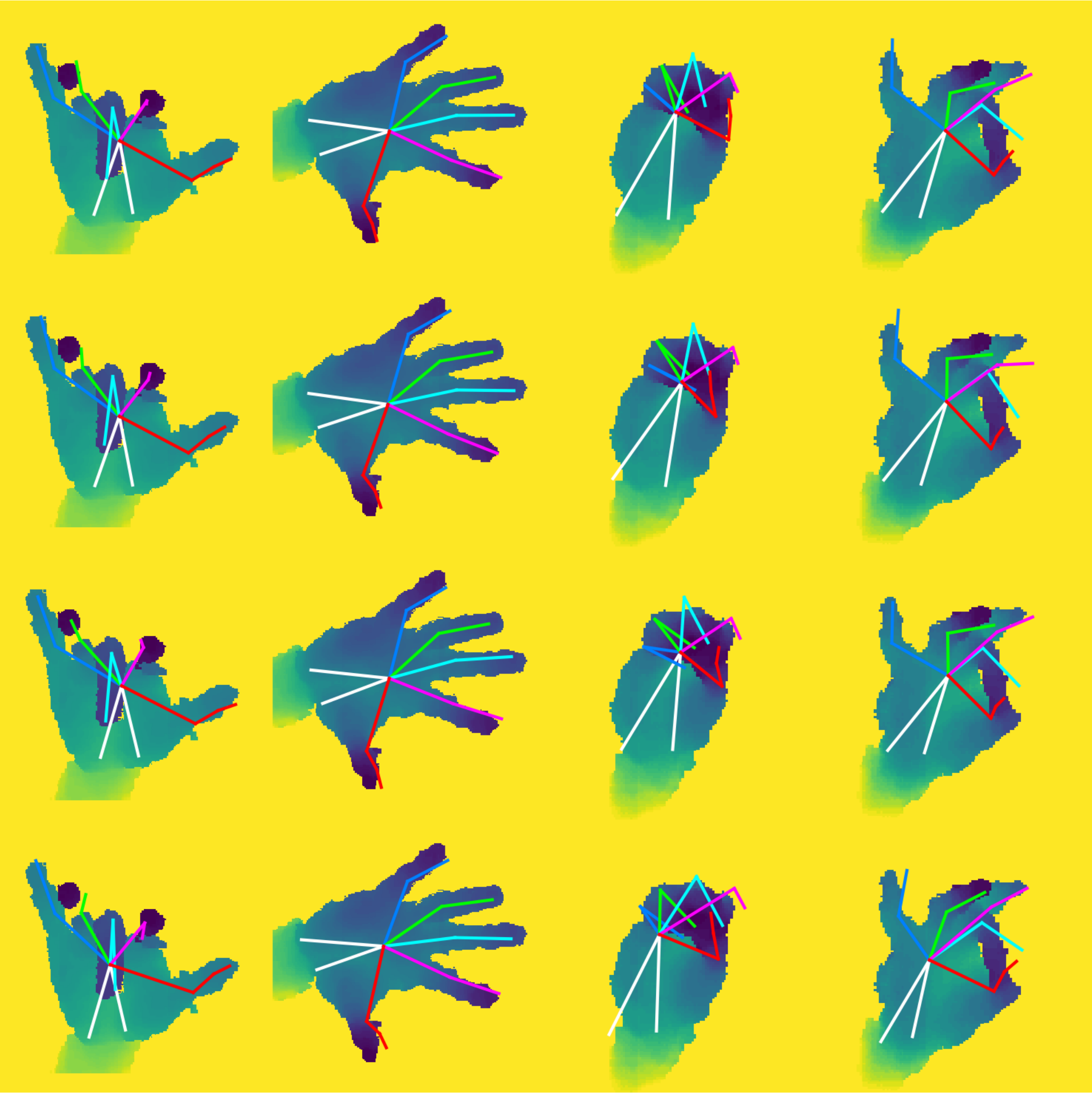
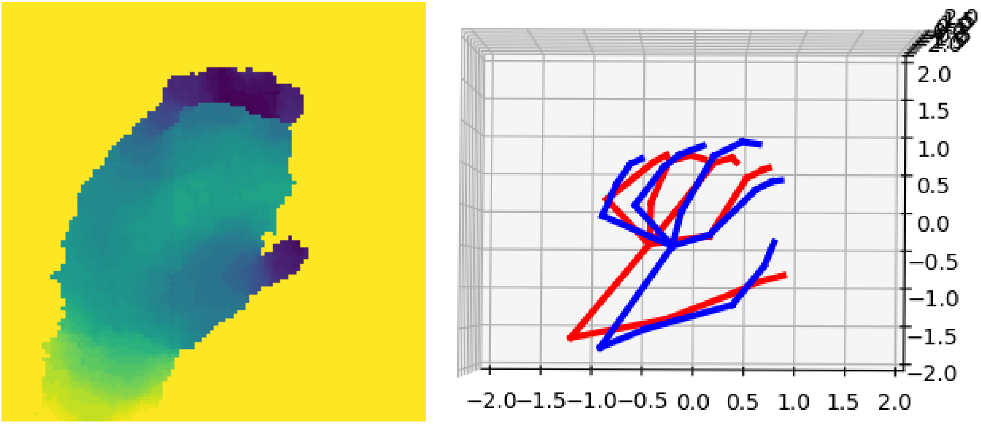
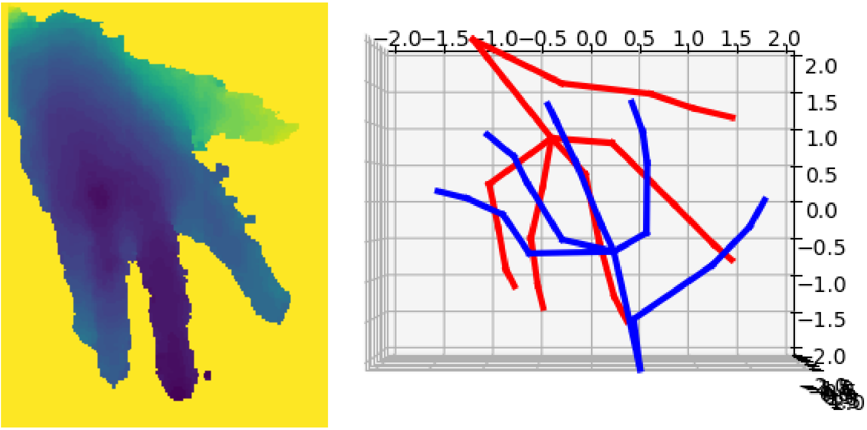


Figure 3.6: Comparison of model predictions as the number of available annotations changes. The top row includes predictions using all annotations. The subsequent rows are predictions from models trained with 75%, 50%, and 25% annotations, respectively.



(A) Input depth image. (B) Prediction (blue) and target (red)

FIGURE 3.7: Reasonable prediction of the target given after training without joint annotations.



(A) Input depth image. (B) Prediction (blue) and target (red)

FIGURE 3.8: The model performs poorly when the hand is seen from uncommon perspectives.

The training time of the Cold Start model was significantly longer than the < 1 of training when fully supervised, converging in 1 week using an NVIDIA Titan V. The model achieved 34.45mm MJE on the NYU test set. The higher error average is mostly due to samples with occlusion and uncommon global orientation.

Figures 3.7 and 3.8 show examples of reasonable accuracy and failure, respectively. Even with augmentation such as rotation, the model is unable to predict uncommon global orientations of the hand.

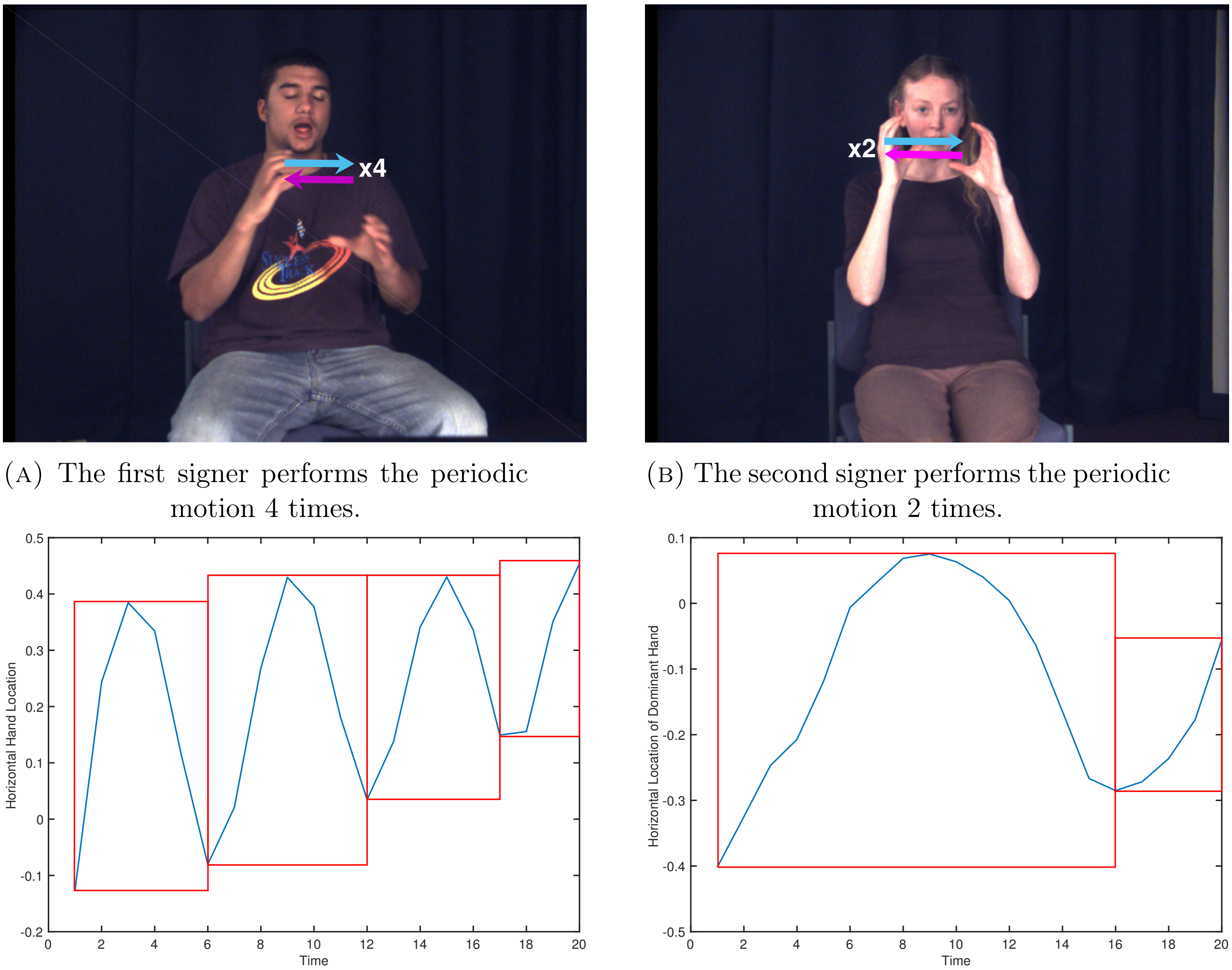
Chapter 4

Detection and Classification of Periodic Sequences in American Sign Language

Sign Language Recognition (SLR) is the task of recognizing the sign or signs in a given video sequence. Input for this task is either considered isolated or continuous. In continuous sign language recognition, the additional challenge of detecting the start and stop frames of each sign is required. For isolated sign language recognition, each input represents a single sign performance. The beginning and end of the sign are assumed to coincide with the video length. An application of this would be an ASL-to-English dictionary system such as the one described in [79].

Multiple forms of variation are inherent in Sign Languages. With respect to isolated sign language recognition, both intra-class and inter-class variations should be considered [11]. Intra-class variation is the variance inherent in performances of the same class by a signer. This can be observed not only through different signers, but of the same signer performing the sign. Inter-class variation refers to the variation between two different signs.

Many of the signs in American Sign Language consist of a single motion and do not include repeated movements. However, there are a significant number of signs that are periodic in nature. A periodic sign includes at least one repeated movement. The inclusion of an additional movement can change the meaning of a sign from its verb form to the related noun. In many cases, a single movement indicates the verb, whereas an additional repeated movement results in the noun. Examples of this include CHAIR/SIT,



(A) The first signer performs the periodic motion 4 times.

(B) The second signer performs the periodic motion 2 times.

(C) 2D plot of dominant hand movement along x-axis over time for the first signer.

(D) 2D plot of dominant hand movement along x-axis over time for the second signer.

FIGURE 4.1: Two different signers performing the sign 'calculus'. The number of additional periods varies between the signers. The horizontal location in the plot is relative to the signer's face. The period (blue) and recovery (magenta) arrows represent the motion of the dominant hand along the horizontal axis.

AIR-PLANE/FLY, and NEWSPAPER/PRINT [74]. There are also certain signs that add repeated movements to indicate the switch from singular to plural [74]. In some cases, the end of a sign movement can be repeated to provide emphasis. Additionally, the number of periods contained in a sign can vary among signers due to personal signing preference. Figure 4.1 shows an example of a signer repeating the original motion three additional times. In these cases, the repeated movements do not change the meaning of the sign.

Large vocabulary sign language datasets often contain few examples per sign. This is due to the large cost of finding expert participants and recording each individual sign

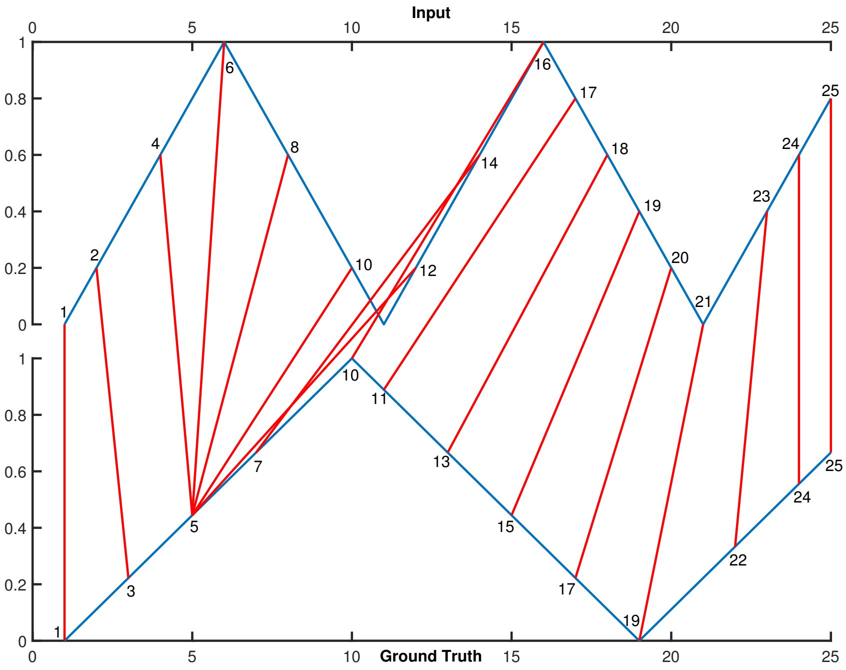


Figure 4.2: Alignment of two similar synthetic examples with different amounts of periodicity. The input sign (top) has an additional period. The red lines are the points of the resulting warping path.

included in the dataset. These limitations result in data that is not well suited for probabilistic or parameterized methods, which require larger amounts of training data. Previous work has focused on the use of Dynamic Time Warping (DTW) [46] for measuring the similarity between two signs. It has shown promising performance for isolated sign language recognition, using as few as one training example per sign class [79].

The measure of similarity from DTW is based directly on the cost of the frame alignment. When comparing two signs in this manner, the resulting alignments are considered meaningful when the inputs contain the same number of periods. This is quantified by a lower alignment cost. Even with frame length normalization, two inputs that represent the same sign could produce a high alignment cost if they differ in the number of repeated movements.

Figure 4.2 shows an alignment resulting from two similar synthetic examples with the input example having one additional period. The blue lines are the 1D time signals. The red lines between the two signals indicate the mapping between individual data points as defined by the warping path. Note that the input (top) subsequence (indices 4 to

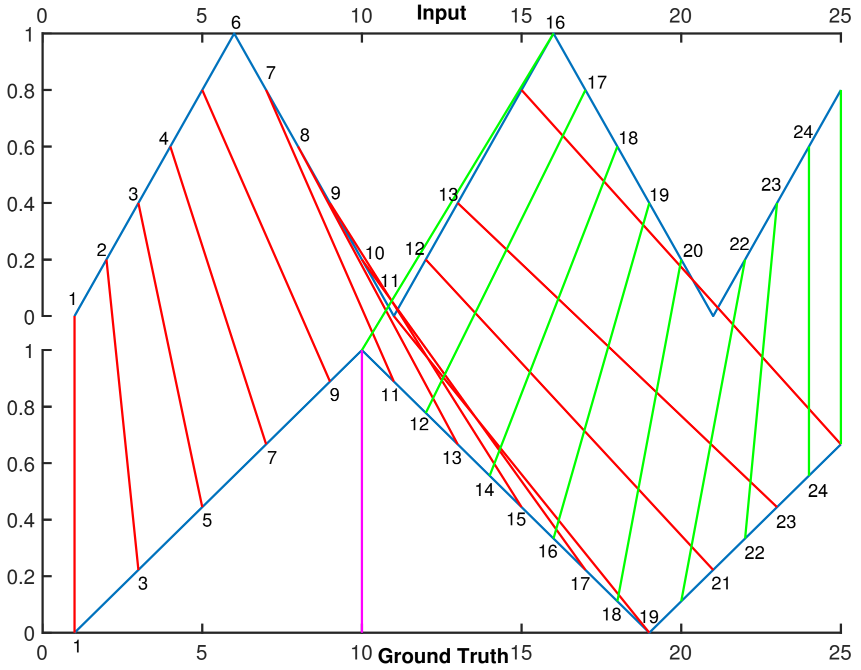


FIGURE 4.3: Meaningful alignment of two similar synthetic examples with differing amounts of periodicity. The input sign (top) has an additional period. This warping path is able to revisit the start of the periodic motion (index 10).

12) is aligned to a single point (index 5) in the ground truth (bottom) sample. For sign language recognition, this alignment would indicate that the two signs are technically dissimilar even if they are semantically the same. Intuitively, an alignment that matches each period in the input to a similar motion in the ground truth is desired. Figure 4.3 shows an example of such an alignment.

In this chapter, two methods based on DTW are evaluated on isolated sign language recognition using periodic data. The first approach uses a newly defined periodic warping path which allows DTW to produce meaningful alignments between periodic sequences with different numbers of periods. The second approach truncates the input sign so that the redundant periodic motion is not evaluated. Both of these approaches assume the start of the periodic sequences of a training sign is known in advance. Truncated DTW additionally requires that the start of the periodic sequence of the test sign is known. They can provide more meaningful alignments in cases of periodic inputs. These changes do not adversely affect the runtime.

The recognition accuracy and quality of alignments of the new methods are evaluated using real sign language data. The results show a small improvement in recognition accuracy over standard DTW, and motivate a need for further research of periodic sign language recognition and automatic periodicity detection methods.

4.1 Related Work

We review some popular and established approaches for the task of Sign Language Recognition. Probabilistic graphical methods such as Hidden Markov Models (HMM) [33, 65, 71, 76] and, by extension, Conditional Random Fields (CRF) [80, 34] have been the most popular. Most of these approaches use an HMM in a Bakis or left-right structure. These structures do not allow transitions to previous states and thus do not explicitly model periodicity.

More recent machine learning methods have also been applied. Fang and Gao use a Recurrent Neural Network (RNN) as a segment detector for the task of continuous SLR [24]. Ong et al. use Sequential Patterns (SP) which provide spatio-temporal feature selection in an efficient tree-based classifier [13]. The drawback to using statistical or machine learning methods is their dependence on larger datasets.

The recent popularity of Deep Learning has motivated new work in SLR. In [54], Pigou et al. employ a Convolutional Neural Network (CNN) as a feature extractor. The feature vectors produced by the CNN are use as input for a neural network based classifier. Koller et al. embed the discriminative power of a CNN into an HMM framework [33]. In this work, the CNN is used to model the emission probability of an HMM.

DTW is an exemplar-based approach that is useful in situations where there is not enough data to train a model. It has been successfully applied in SLR as both a classifier [69, 79] and as a distance measure for extracting the most similar segment between multiple sign sentences [47]. Our proposed method evaluates the effect of periodicity in a DTW-based SLR system. We refer the reader to a survey by Cooper et al. for more information on SLR [12].

Previous works have evaluated periodic motions. He et al. use HMMs for the task of periodic activity recognition [32]. The structure includes a transition from the last state of the HMM to the beginning. Ruiz et al. show that RNNs can replicate a time varying periodic signal [59]. Our proposed approach is motivated by that of an earlier study of periodic SLR [51].

4.2 Method

We propose an analytical method for comparing two signals in which one may have a varying number of similar periodic subsequences. Our method is a relaxed definition of DTW, thus we call it Periodic DTW. We provide a brief overview of DTW and its definition followed by a description of Periodic DTW.

4.2.1 Dynamic Time Warping

Dynamic Time Warping measures the similarity of two temporal sequences. One of the many benefits of DTW is that it is robust to differences in the speed and length of the sequence. This is important for tasks such as sign language recognition because the speed at which a sign is performed can vary between users. DTW produces a warping path which serves as an alignment between the inputs in the time dimension. The cost of aligning two inputs using DTW provides a reliable similarity measure which can be useful for classification tasks.

We follow the description and notation from [46]. Given two sign inputs $X = (x_1, x_2, \dots, x_N)$ and $Y = (y_1, y_2, \dots, y_M)$, DTW computes a warping path $W = (w_1, \dots, w_L)$ of length L where $w_l = (n_l, m_l)$ refers to the mapping from frame X_{n_l} to frame Y_{m_l}. In other words, W provides an alignment between X and Y. The warping path must satisfy the following three constraints: boundary, monotonicity, and step size. The boundary constraint ensures that the first and last frames of X are aligned to the first and last frames of Y. The step size and monotonicity constraints restrict the warping path from skipping frames or jumping backwards in time.

$$\begin{aligned} \textbf{boundary:}\ & w_1 = (1,1) \text{ and } w_L = (N, M) \\ \textbf{monotonicity:}\ & n_1 \leq n_2 \leq \cdots \leq n_L \text{ and} \\ \textbf{step size:}\ & w_{l+1} - w_l \in (1,0), (0,1), (1,1) \\ & \text{for } l \in [1 : L-1]. \end{aligned}$$

In sign language recognition, the alignment cost provided by DTW is used for classifying a sign by selecting the lowest cost comparison between all signs in the dictionary. The cost $C(W, X, Y)$ of a warping path is defined as the sum of the local costs corresponding to the alignment of X and Y:

$$C(W, X, Y) = \sum_{l=1}^{L} c(X_{n_l}, Y_{m_l}). \tag{4.1}$$

The local cost $C(x_{n_l}, Y_{m_l})$ can be defined as the Euclidean distance between the feature vectors describing each frame $X_{n_l} and Y_{m_l}$. DTW calculates the overall lowest cost provided by all possible warping paths:

$$DTW(X, Y) = \min_{W} C(W, X, Y). \tag{4.2}$$

The cost of alignment produced by $DTW(X, Y)$ is then evaluated for all signs $Y \in \mathbb{Y}$ in the training set. The sign recognized by the system is the label of the sign Y corresponding to the lowest cost returned by $DTW(X, Y)$.

4.2.2 Periodic Warping Path

The standard definition of a warping path is too restrictive to produce a warping path between signals with periodicity. As a result, the higher cost of alignment may lead to misclassifications in a sign language recognition system. If the warping path was relaxed such that periodic movements could be revisited, DTW could produce alignments between data points in the test sign with those that are semantically similar in the training sign.

Following [51], we loosely define a periodic sign by its recovery and period movements. The period movement is defined as the motion required by the signer to gesture the sign. The recovery movement is the motion of returning from the end of the period movement back to the beginning of it. Figure 4.1 shows two signers performing the sign 'calculus'. The period (blue) and recovery (magenta) motions along the horizontal axis are overlaid onto the image.

Towards classifiying periodic signs, we define a periodic warping path that allows DTW to revisit the start of a periodic movement. Let r be the frame of the start of the recovery motion of a sign. The periodic warping path $W = (w_1, \ldots, w_L)$ satisfies the following constraints:

$$\begin{aligned} \textbf{boundary:}\ & w_1 = (1,1) \text{ and } w_L = (N, M) \\ \textbf{montonicity:}\ & n_1 \leq n_2 \leq \cdots \leq n_L \\ \textbf{step size:}\ & w_{l+1} - w_l \in (1,0), (0,1), (1,1), (0, r - m_l) \\ & \text{for } l \in [l : L-1], m_l > r. \end{aligned}$$

Note that the step size constraint is the only real change between a standard and periodic warping path. The change in the monotonicity constraint is implied by the step size constraint. In practice, DTW can now map multiple frames of the test sign X to that of the recovery start frame in the training sign Y. No other changes need to be made for DTW to provide an alignment using the periodic warping path.

Using the recovery frame r, a periodic warping path can transition to the beginning of a periodic motion at any point $m_l > r$. An example of this is shown in Figure 4.4. There is no restriction on the number of times Periodic DTW can map back to r.

Using this new definition allows DTW to generate warping paths that were not possible under the standard definition. However, this can lead to misclassifications as well. The looser constraints of a periodic warping path may not always produce the desired result.

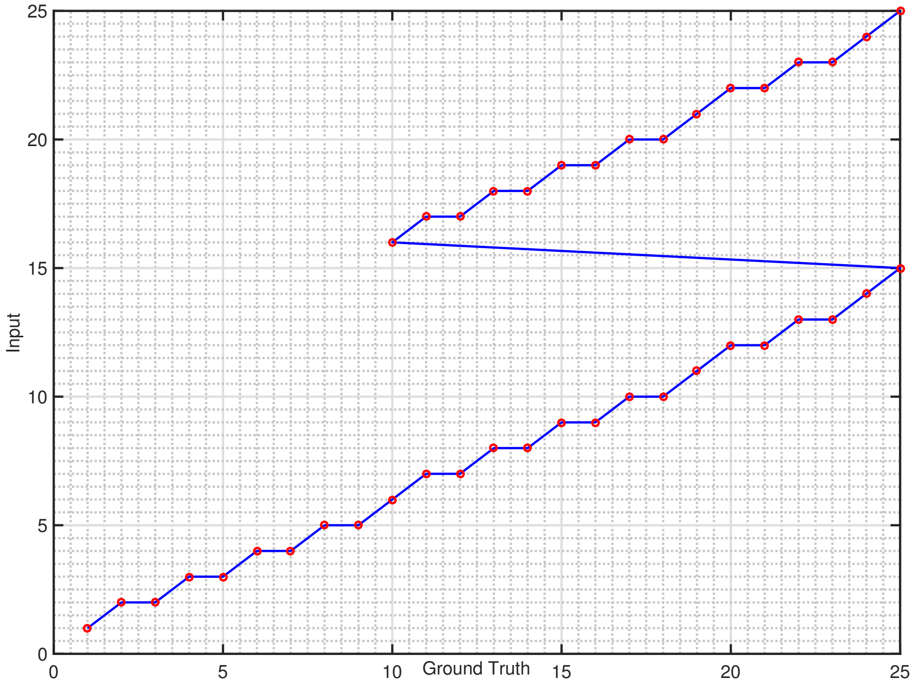


FIGURE 4.4: Visualization of the periodic warping path as seen in Figure 4.3. The input aligns to the end of the ground truth and then jumps back to recovery frame 10.

Thus, a periodic warping path is not a perfect solution to the problem. Several properties of the data can lead to misclassifications. The periodic subsequences are not uniform, and noise from signing, image capture, and other factors can cause these subsequences to vary with respect to a sign. An example of this is shown in 4.1.

4.3 Experiments

We compare the proposed modification to DTW, periodic DTW, using sequential keypoints data from the ASLLVD dataset [4]. This dataset features 1,113 distinct sign classes. There are three examples performed by independent singers for each sign class. We focus only on examples that are periodic in nature and consist of a different number of periods between each signer. Following the definitions introduced in [51], we further classify periodic signs as periodic, pure, and/or non-circular. A periodic and pure sign is one that has a similar trajectory path from one occurrence to the next. A non-circular sign is one that has no circular motions. Between the two subsets used, there are 207 signs that are periodic, pure, and non-circular with a differing number of periods. We use all of these examples to build the test set.

The manual periodic annotations are provided by earlier work on periodic sign language recognition [51]. We define signs with excess periodic movements as those with more than two periods. As stated previously, repeating a subsequence for a particular sign can change the actual meaning of it, but repetitions in excess of two periods are merely added for emphasis.

4.3.1 Evaluation Protocol

To evaluate the performance of each method, we follow the accuracy measures described in [79]. That is, given a query sign X, the measure of performance is the rank $R(X)$ that the method assigns to the correct result for X. Given an integer k, we use a Boolean measure of success $S(X, k)$, that is true if and only if $R(X) \leq k$. The success rate $S(k)$ over a test set of queries is the average success rate $S(X, k)$ over the test set.

4.3.2 Features and Normalization

Following the description given in [79], we extract location and orientation features from each frame of a sign video. The features are derived from the locations of the hands from each frame. For these experiments, we use manual annotations provided by [51] to minimize the amount of input noise.

Given a sign video X of length N, let X_n denote the n-th frame of the video, where $n \in [1 : N]$. The features derived from X_n are as follows:

- $L_d(X_n)$ and $L_{nd}(X_n)$: The (x, y) centroid corresponding to the dominant hand and non-dominant hand, respectively, of the signer at frame n.

- $L_\delta(X_n)$: The relative position of the dominant hand with respect to the non-dominant hand at frame n. $L_\delta(X_n) = L_d(X_n) - L_{nd}(X_n)$.

- $O_d(X_n)$ and $O_{nd}(X_n)$: The unit vectors representing the direction of motion from $L_d(X_{n-1})$ to $L_d(X_{n+1})$ and from $L_{nd}(X_{n-1})$ to $L_{nd}(X_{n+1})$.
- $O_\delta(X_n)$: The unit vector representing the direction of motion from $L_\delta(X_{n-1})$ to $L_\delta(X_{n+1})$.

For these experiments, we do not use any hand appearance features.

For one-handed signs, the non-dominant hand features L_{nd}, L_δ, O_{nd}, and O_δ are not calculated. Instead, these features are set to 0 for each frame of the input. All signs are resampled using linear interpolation and have a length of 20 frames for these experiments. The hand locations are normalized with respect to the diagonal length of a bounding box containing the face. This normalization is necessary due to the variation in the size of the person signing and their distance from the camera.

Each feature differs in their discriminative capabilities. The range of values is also different between each one. For these reasons, a weighted local cost function is used to calculate the alignment cost for DTW. In our implementation, the weighting is done during feature processing. Given the features defined above, the local cost function used is defined as follows:

$$\begin{aligned} c(X_{n_l}, Y_{m_l}) = & f_1 \| L_d(X_{n_l}) - L_d(Y_{m_l}) \| + \\ & f_2 \| L_{nd}(X_{n_l}) - L_{nd}(Y_{m_l}) \| + \\ & f_3 \| L_\delta(X_{n_l}) - L_\delta(Y_{m_l}) \| + \\ & f_4 \| O_d(X_{n_l}) - O_d(Y_{m_l}) \| + \\ & f_5 \| O_{nd}(X_{n_l}) - O_{nd}(Y_{m_l}) \| + \\ & f_6 \| O_\delta(X_{n_l}) - O_\delta(Y_{m_l}) \| . \end{aligned}$$

4.4 Alignment Visualization

Besides comparing the overall recognition accuracy of these methods, it is important to look at the quality of the alignment provided in each case. By looking at the resulting alignment between two sign inputs with a differing number of periods, we can easily observe how standard DTW is not well suited for periodic signs. Figure 4.5 shows the alignments provided by each of the described methods. In the figure, the test sample is shown on the top while the training sign is shown on the bottom. The red lines linking the two examples indicate the alignment provided by DTW. Some of the lines were removed from the figures for clarity.

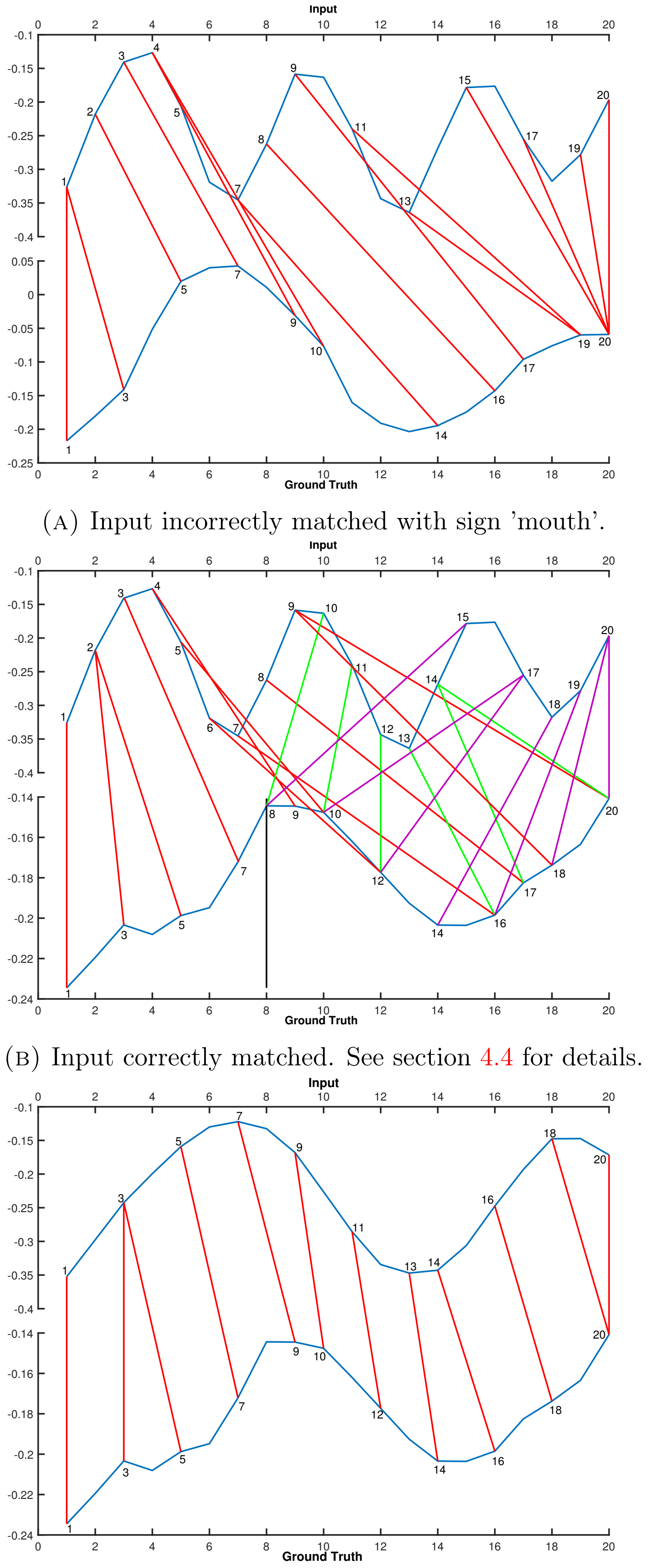
In Figure 4.5b, the green and megenta lines are those that are mapped as a result of the periodic warping path. Note, for example, that point 10 in the input (top) example aligns back to point 8 in the ground truth (bottom) example. This example shows how the periodic warping path matches the subsequences in the input properly when compared to classical DTW. The period in the input sign from frame 10 to frame 14 matches the shape in the ground truth from frame 8 to frame 20. Likewise, the magenta alignment from input frames 15 to 20 matches the shape of the ground truth from frames 8 to 20. The black vertical line at frame 8 indicates the recovery start frame r.

4.5 Results

The results of the classification experiment described in the previous section 4.3 are reported here. We compare standard DTW and DTW with periodic warping paths. Additionally, we use a separate set of data in which the periodic motions are truncated to examine the case in which redundant motions are omitted.

We plot the top-k classification accuracy of the three DTW-based methods in Figure 4.6. The final accuracy scores are averaged between the result of using test samples from TB and training from LB and vice versa. The percentage of queries for which the correct sign is in the top 10 results if 55% for standard DTW, 57% for DTW with periodic alignments, and 64% using DTW with truncated inputs.

Using periodic warping paths shows a small improvement over standard DTW in our experiments. Individual comparisons exemplify the ability of periodic warping paths to provide a lower cost alignment. The alignments provided by this approach match



(A) Input incorrectly matched with sign 'mouth'.

(B) Input correctly matched. See section 4.4 for details.

(C) Input correctly matched.

Figure 4.5: The resulting alignment using sign 'beer' as input from (a) DTW, (b) DTW with a periodic warping path, and (c) DTW with truncated inputs.

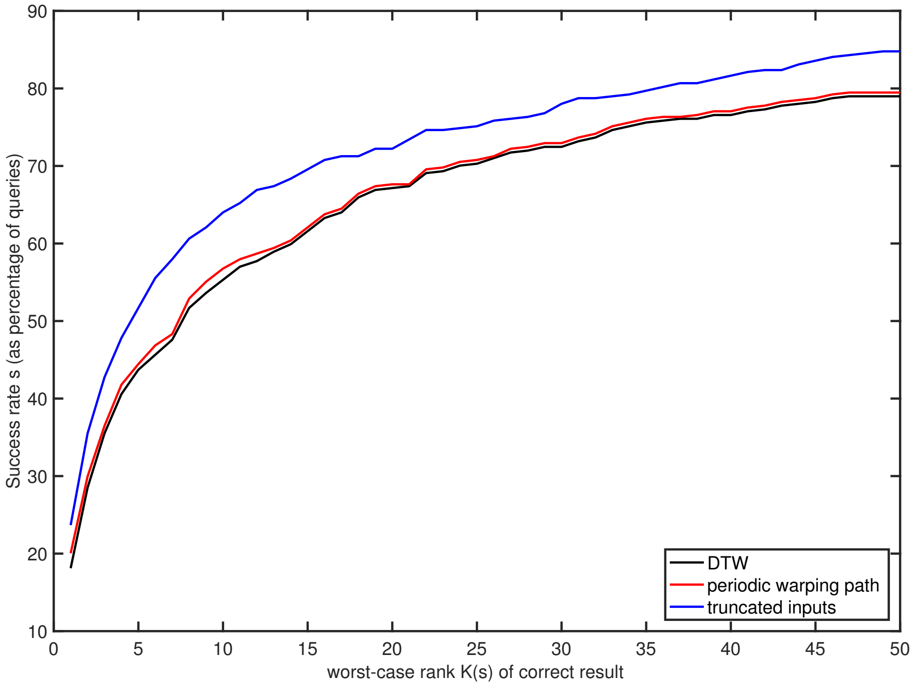


FIGURE 4.6: Comparison of DTW with a standard warping path, periodic warping path, and truncated inputs. The results are cross-validated between the TB and LB datasets. The x-axis corresponds to values of $K(s)$.

the periodic subsequences in a meaningful way. That is, the periodic subsequences are matched to a similar shape in the ground truth.

The downside to this approach is the ability to provide additional warping paths that were not previously possible under the standard definition of DTW. A practical result of this would be a lower cost alignment for signs that are semantically different. An example of this is shown in Figure 4.7.

The results show a stronger case for using truncated inputs with standard DTW. In both experiments, the overall results were better using DTW with truncated inputs than with DTW using periodic warping paths. The first benefit to using truncated inputs is that the standard definition of a warping path can be used. These tightened constraints prevent erroneous alignments as described previously. The main benefit of using truncated inputs is that the redundant periods are no longer considered as part of the alignment. The periodic movements signed are typically not rigid and can vary between periods of the same sign.

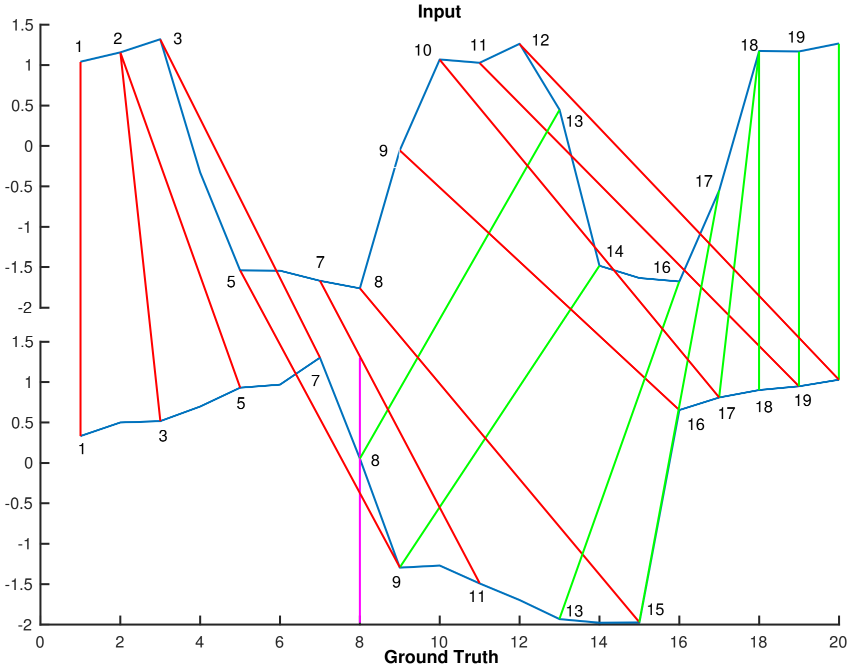


FIGURE 4.7: Two semantically different signs are matched using a periodic warping path. The input sign is 'aunt' and the ground truth sign is 'history'. The red and green lines represent the alignment. The green lines are those that are mapped due to the periodic warping path.

Chapter 5

Conclusions and Future Work

In this dissertation we evaluated challenging problems related to human motion analysis. We first introduced a novel method for handling a special case of signs from American Sign Language. Toward a fully automated assessment and training tool for cognitive development, we introduced a data collection framework resulting in a novel dataset for the Activate Test of Embodied Cognition. Using real data from this evaluation, we adapted established machine learning methods to provide automatic scores for selected tasks within the ATEC suite. Finally, this dissertation explored self-supervised learning to address the burden of manual annotations and a multitude of untapped, yet unlabelled, data.

We conclude this dissertation with a brief review of each chapter, summarizing the work and contributions therein. Discussions of future work are included as these challenges are ongoing and require further investigation before robust and accessible solutions can be provided.

5.1 Periodic Sign Language Recognition

We proposed a novel method for handling periodic signs in American Sign Language. Three separate approaches based on Dynamic Time Warping were evaluated. The first used a standard definition of a warping path. The warping path was then modified to allow periodic warping paths. The final approach simulated the performance of the first method if the periodic subsequences were truncated. For each method, we analyzed two desirable properties: recognition accuracy and the quality of the resulting alignments.

The results of these experiments show a clear improvement in recognition accuracy when the system can properly handle periodic inputs. Using periodic warping paths produced more meaningful alignments which led to a marginal increase in recognition accuracy in our tests on the ASLLVD dataset. However, the relaxed constraints can lead to warping paths that were not previously possible under the standard definition. This could lead to new misclassifications in a sign language recognition system. For example, two signs that are semantically different could be matched incorrectly. Using truncated inputs with a standard definition of Dynamic Time Warping produced the greatest accuracy improvement in these experiments.

In both the case of periodic warping paths and truncated inputs, the start of the recovery frames is provided by manual annotation. For periodic warping paths, the start of the recovery frames only needs to be provided for the training examples. If we use truncated inputs, the start of the recovery frames also should be known for the test signs. Providing such manual annotations for large datasets is not always feasible and motivates the need for an automatic method. Furthermore, requiring such information to be provided for test signs makes the user interface more cumbersome. Future work will look into detecting the subsequences of a sign to detect recovery periods in an automatic way. The output of these detected recovery periods could be used in place of the manual annotations used in this work.

5.2 Activate Test of Embodied Cognition

The main research goal is to design a fully automated, high fidelity, and low-cost assessment system for embodied cognition. This dissertation contributes toward this goal by introducing the data collection framework used to collect all data related to the Activate Test of Embodied Cognition as well as baseline evaluations for selected tasks. Preliminary results were reported towards an automated scoring approach for three core tasks of the Activate Test for Embodied Cognition: *Ball Drop*, *Sailor Song*, and *Finger Tap*.

As methods for automatically scoring each task are introduced, a fully automated system for the assessment and training of certain cognitive behaviors can be achieved. Such a system could reduce the societal and economic burdens associated with untreated cognitive disorders.

This work is ongoing and will include more data collection and improvement of the methods used to automatically score each task. Further analysis can be used to extract more information related to performance delays and speed, self-correction, and extraneous movements, which will aid future research in identified and modeling individual differences in child performance.

5.3 Self-Supervised Hand Pose Estimation

Fully supervised training is currently the most efficient ways to train deep neural networks. However, the cost of annotating data can be expensive or even impossible, especially when considering novel tasks and datasets. Ignoring unlabelled samples is a waste of valuable data that can be repurposed using self-supervision.

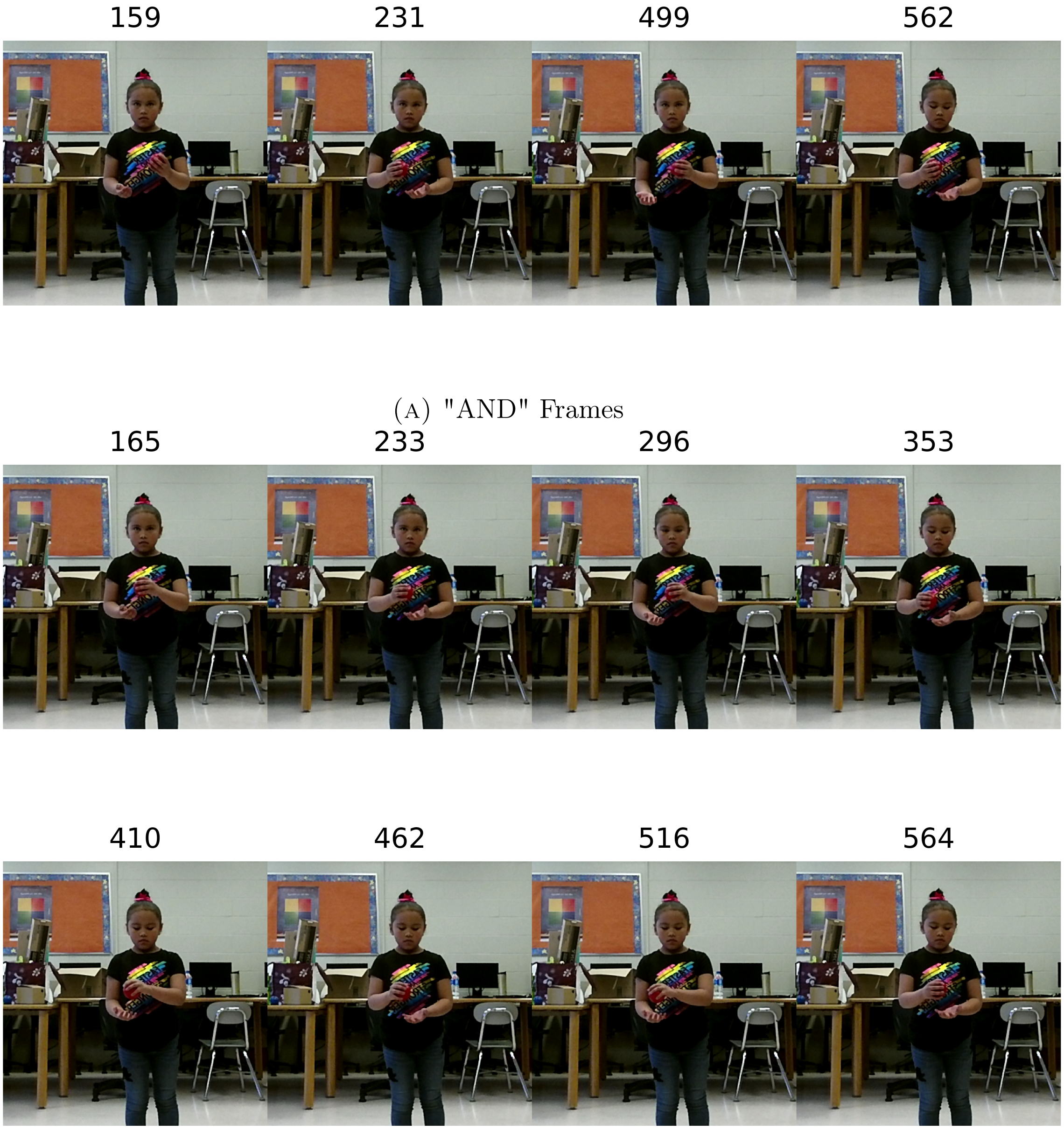
Self-supervised Hand Pose Estimation was explored as part of this dissertation. It can be integrated into existing models in a number of ways, and we explore two cases. First, we analyze the effect of training a hand pose estimation model as the number of labelled samples are reduced. This is a more realistic case as it may be feasible to provide a small amount of annotations for a particular dataset while allowing the model to interpolate the remaining cases. We evaluate our framework on a popular hand pose estimation dataset [73] and achieve comparative results with other self-supervised approaches.

Additionally, a more extreme case of missing labelled data is explored which we refer to as training from a Cold Start. In this case, a model is initialized with no labelled data as well. Ideally, a general learning framework would generalize the specific representations of the data without explicit supervision. Besides furthering our understanding of deep models and the representations they encapsulate, such a scenario would increase the accessibility of deep learning to novel tasks and datasets. We evaluate this scenario using additional modes of self-supervision on multi-view data.

Appendix A

Bag Pass Detected Frames

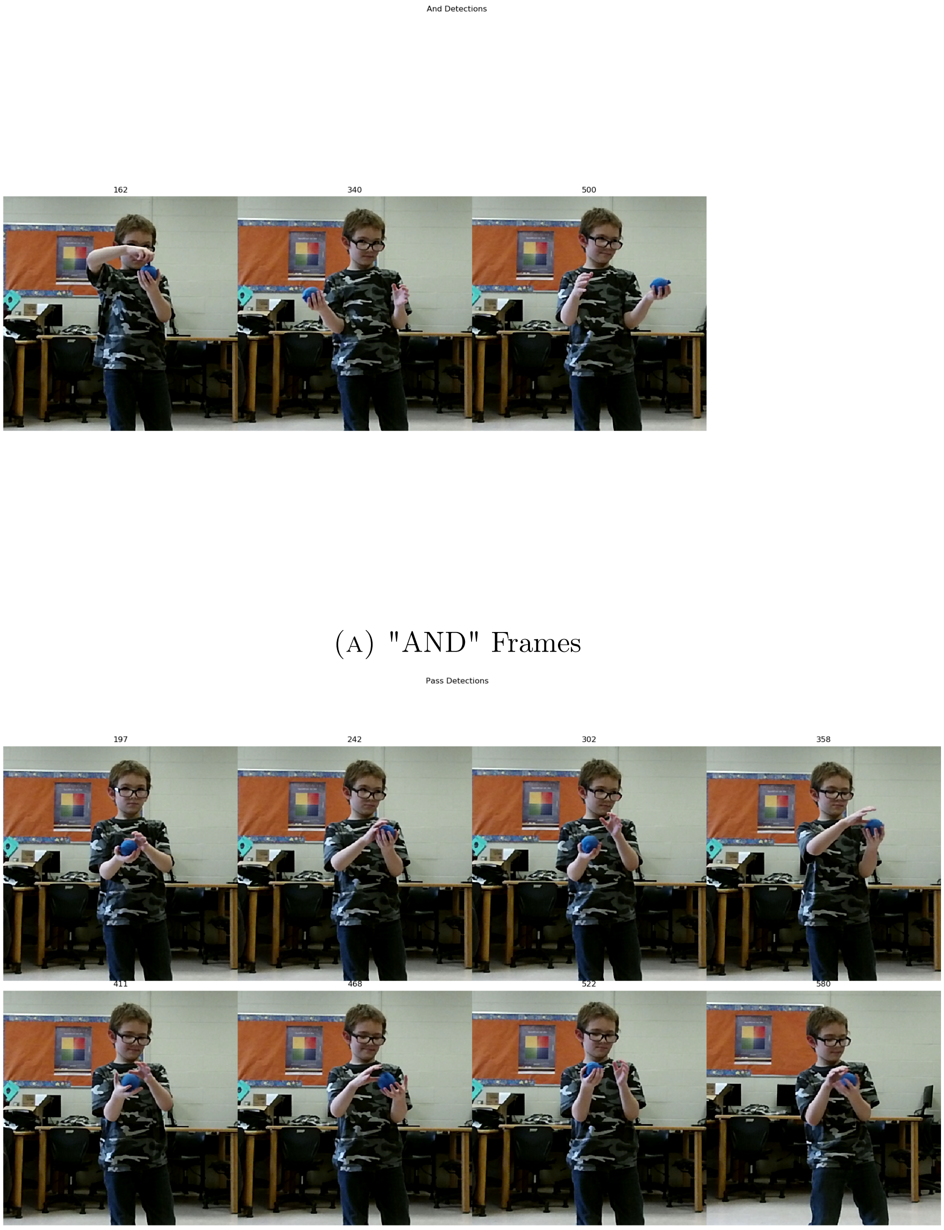
The frames at which the Viterbi-style approach detected an "AND" or "PASS" movement are important not only for considering improvements to future methods, but for analyzing extraneous movements or unexpected behaviors of each subject.



(A) "AND" Frames

(B) "PASS" Frames

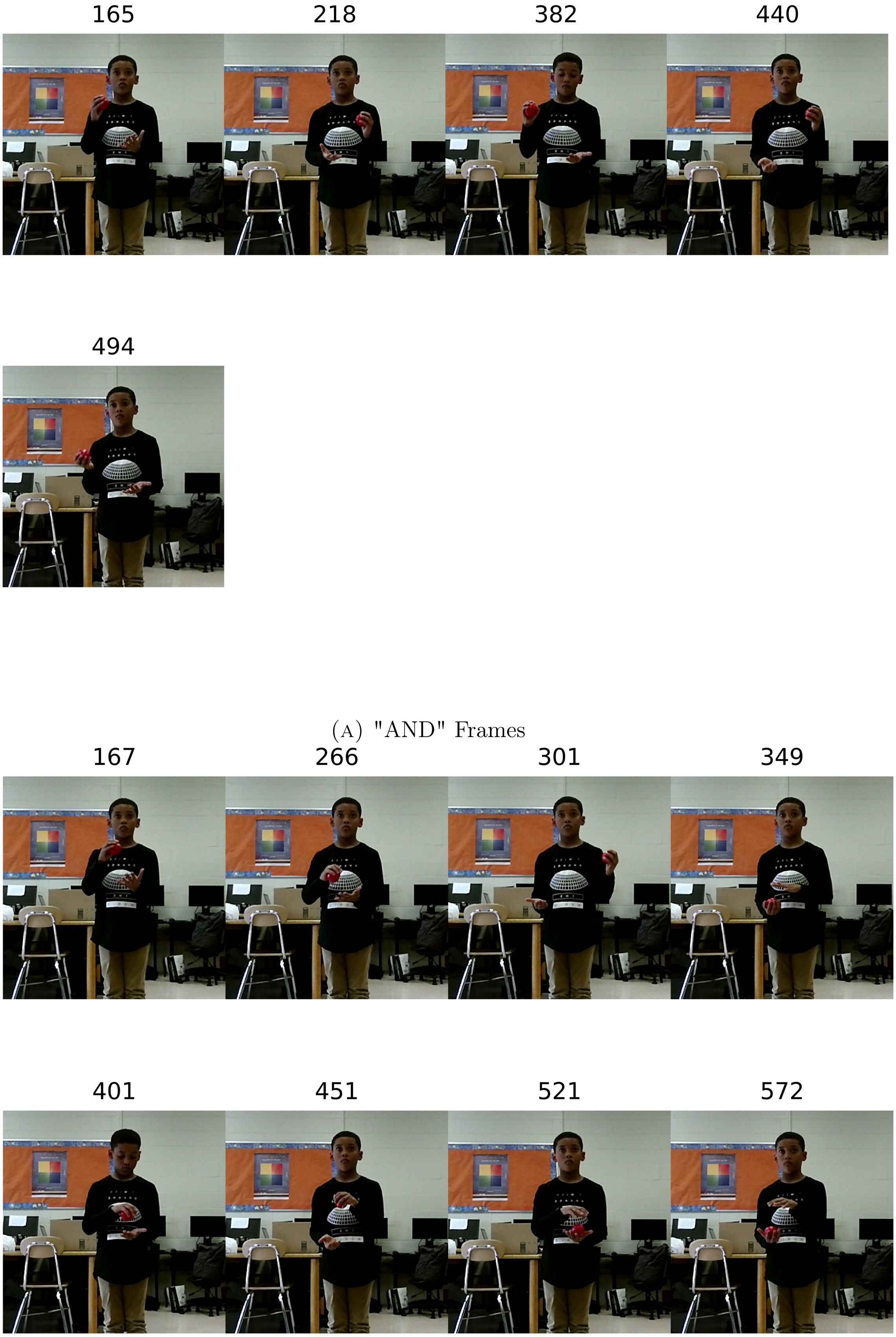
FIGURE A.1: Detected frames corresponding to the "AND" (top) and "PASS" (bottom) movements for Subject 1.



(A) "AND" Frames

(B) "PASS" Frames

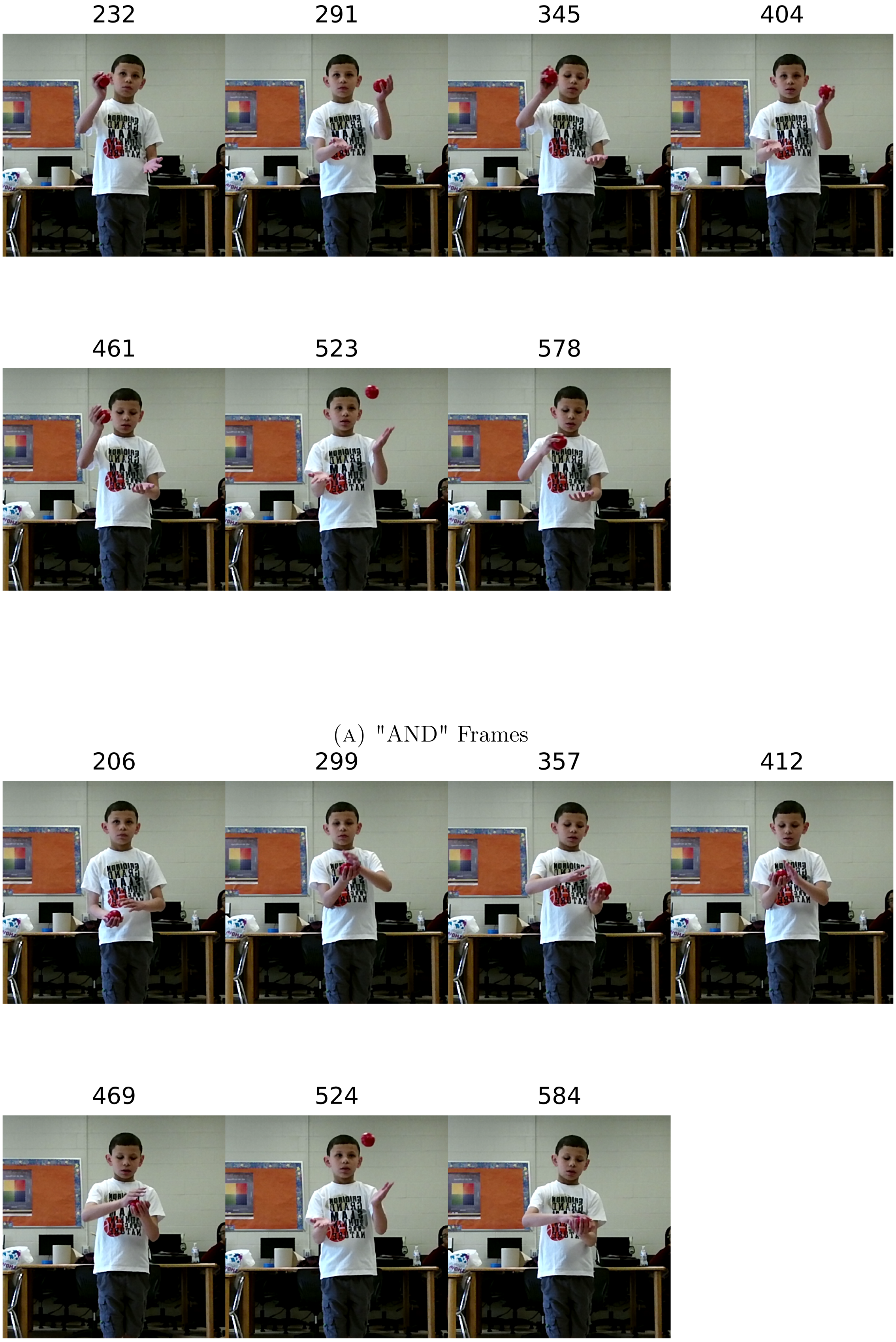
FIGURE A.2: Detected frames corresponding to the "AND" (top) and "PASS" (bottom) movements for Subject 2.



(A) "AND" Frames

(B) "PASS" Frames

FIGURE A.3: Detected frames corresponding to the "AND" (top) and "PASS" (bottom) movements for Subject 3.



(A) "AND" Frames

(B) "PASS" Frames

FIGURE A.4: Detected frames corresponding to the "AND" (top) and "PASS" (bottom) movements for Subject 4.

Bibliography

[1] Masoud Abdi et al. “3D Hand Pose Estimation using Simulation and Partial-Supervision with a Shared Latent Space”. In: ().

[2] Thomas M Achenbach, Thomas M Ruffle, et al. “The Child Behavior Checklist and related forms for assessing behavioral/emotional problems and competencies”. In: *Pediatrics in review* 21.8 (2000), pp. 265–271.

[3] Alaaeldin Ali and Graham W Taylor. “Real-Time End-to-End Action Detection with Two-Stream Networks”. In: *2018 15th Conference on Computer and Robot Vision (CRV)*. IEEE. 2018, pp. 31–38.

[4] Vassilis Athitsos et al. “The American Sign Language Lexicon Video Dataset”. In: *2008 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops, CVPR Workshops*. ISBN: 9781424423408. DOI: 10.1109/CVPRW.2008.4563181.

[5] Russell A Barkley et al. “ADHD symptoms vs. impairment: revisited”. In: *The ADHD Report: Special Issue—Focus on Assessment* 14.2 (2006), pp. 1–9.

[6] Russell A Barkley et al. “Young adult outcome of hyperactive children: adaptive functioning in major life activities”. In: *Journal of the American Academy of Child & Adolescent Psychiatry* 45.2 (2006), pp. 192–202.

[7] Benjamin Buchanan, Konstantinos Tsiakas, and Morris Bell. “Towards an automated assessment for embodied cognition in children: the sailor step task”. In: *Proceedings of the 12th ACM International Conference on PErvasive Technologies Related to Assistive Environments*. ACM. 2019, pp. 331–332.

[8] Yujun Cai et al. “Weakly-supervised 3d hand pose estimation from monocular rgb images”. In: *Proceedings of the European Conference on Computer Vision (ECCV)*. 2018, pp. 666–682.

[9] Zhe Cao et al. “OpenPose: realtime multi-person 2D pose estimation using Part Affinity Fields”. In: *arXiv preprint arXiv:1812.08008*. 2018.

[10] Yujin Chen et al. “So-handnet: Self-organizing network for 3d hand pose estimation with semi-supervised learning”. In: *Proceedings of the IEEE International Conference on Computer Vision*. 2019, pp. 6961–6970.

[11] C. Conly, A. Dillhoff, and V. Athitsos. “Leveraging intra-class variations to improve large vocabulary gesture recognition”. In: *2016 23rd International Conference on Pattern Recognition (ICPR)*. 2016, pp. 907–912.

[12] Helen Cooper, Brian Holt, and Richard Bowden. “Sign language recognition”. In: *Visual Analysis of Humans*. Springer, 2011, pp. 539–562.

[13] Helen Cooper et al. “Sign language recognition using sub-units”. In: *Journal of Machine Learning Research* 13.Jul (2012), pp. 2205–2231.

[14] Catherine L Davis and Stephanie Cooper. “Fitness, fatness, cognition, behavior, and academic achievement among overweight children: do cross-sectional associations correspond to exercise trial outcomes?” In: *Preventive medicine* 52 (2011), S65–S69.

[15] Emma E Davis, Nicola J Pitchford, and Ellie Limback. “The interrelation between cognitive and motor development in typically developing children aged 4–11 years is underpinned by visual processing and fine manual control”. In: *British Journal of Psychology* 102.3 (2011), pp. 569–584.

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

[17] Xiaoming Deng et al. “Hand3d: Hand pose estimation using 3d neural network”. In: *arXiv preprint arXiv:1704.02224* (2017).

[18] Maxime Devanne et al. "3-d human action recognition by shape analysis of motion trajectories on riemannian manifold". In: *IEEE transactions on cybernetics* 45.7 (2014), pp. 1340–1352.

[19] Endri Dibra et al. "How to refine 3d hand pose estimation from unlabelled depth data?" In: *2017 International Conference on 3D Vision (3DV)*. IEEE. 2017, pp. 135–144.

[20] Endri Dibra et al. "Monocular RGB hand pose inference from unsupervised refinable nets". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*. 2018, pp. 1075–1085.

[21] Joseph E Donnelly and Kate Lambourne. "Classroom-based physical activity, cognition, and academic achievement". In: *Preventive medicine* 52 (2011), S36–S42.

[22] Jalpa A Doshi et al. "Economic impact of childhood and adult attention-deficit/hyperactivity disorder in the United States". In: *Journal of the American Academy of Child & Adolescent Psychiatry* 51.10 (2012), pp. 990–1002.

[23] Ali Erol et al. "Vision-based hand pose estimation: A review". In: *Computer Vision and Image Understanding* 108.1-2 (2007), pp. 52–73.

[24] Gaolin Fang and Wen Gao. "A SRN/HMM system for signer-independent continuous sign language recognition". In: *Proceedings - 5th IEEE International Conference on Automatic Face Gesture Recognition, FGR 2002* (), pp. 312–317. DOI: 10.1109/AFGR.2002.1004172.

[25] Hao-Shu Fang et al. "Rmpe: Regional multi-person pose estimation". In: *Proceedings of the IEEE International Conference on Computer Vision*. 2017, pp. 2334–2343.

[26] Jens Forster et al. "RWTH-PHOENIX-Weather: A Large Vocabulary Sign Language Recognition and Translation Corpus." In: *LREC*. 2012, pp. 3785–3789.

[27] Liuhao Ge et al. "Robust 3d hand pose estimation in single depth images: from single-view cnn to multi-view cnns". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 3593–3601.

[28] Ian Goodfellow et al. "Generative adversarial nets". In: *Advances in neural information processing systems*. 2014, pp. 2672–2680.

[29] Andrea L Green and David L Rabiner. "What do we really know about ADHD in college students?" In: *Neurotherapeutics* 9.3 (2012), pp. 559–568.

[30] Hengkai Guo et al. "Region ensemble network: Improving convolutional network for hand pose estimation". In: *2017 IEEE International Conference on Image Processing (ICIP)*. IEEE. 2017, pp. 4512–4516.

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

[32] Qiang He and Christian Debrunner. "Individual recognition from periodic activity using hidden markov models". In: *Human Motion, 2000. Proceedings. Workshop on.* IEEE. 2000, pp. 47–52.

[33] Oscar Koller et al. "Deep Sign: Hybrid CNN-HMM for Continuous Sign Language Recognition". In: *Proceedings of the British Machine Vision Conference 2016*. 2016.

[34] W. W. Kong and Surendra Ranganath. "Towards subject independent continuous sign language recognition: A segment and merge approach". In: *Pattern Recognition* 3 (), pp. 1294–1308. ISSN: 00313203. DOI: 10.1016/j.patcog.2013.09.014.

[35] Hildegard Kuehne et al. "HMDB: a large video database for human motion recognition". In: *2011 International Conference on Computer Vision*. IEEE. 2011, pp. 2556–2563.

[36] Chenyang Li et al. "Skeleton-based Gesture Recognition Using Several Fully Connected Layers with Path Signature Features and Temporal Transformer Module". In: *arXiv preprint arXiv:1811.07081* (2018).

[37] Jiaxin Li, Ben M Chen, and Gim Hee Lee. "So-net: Self-organizing network for point cloud analysis". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018, pp. 9397–9406.

[38] Mehran Maghoumi and Joseph J LaViola Jr. “DeepGRU: Deep gesture recognition utility”. In: *International Symposium on Visual Computing*. Springer. 2019, pp. 16–31.

[39] Jameel Malik, Ahmed Elhayek, and Didier Stricker. “Simultaneous hand pose and skeleton bone-lengths estimation from a single depth image”. In: *2017 International Conference on 3D Vision (3DV)*. IEEE. 2017, pp. 557–565.

[40] Jameel Malik et al. “Deephps: End-to-end estimation of 3d hand pose and shape by learning from synthetic depth”. In: *2018 International Conference on 3D Vision (3DV)*. IEEE. 2018, pp. 110–119.

[41] Robert M Malina, Sean P Cumming, and Manuel J Coelho-e Silva. “Physical Activity and Inactivity Among Children and Adolescents: Assessment, Trends, and Correlates”. In: *Biological Measures of Human Experience across the Lifespan*. Springer, 2016, pp. 67–101.

[42] Marcin Marszałek, Ivan Laptev, and Cordelia Schmid. “Actions in context”. In: *CVPR 2009-IEEE Conference on Computer Vision & Pattern Recognition*. IEEE Computer Society. 2009, pp. 2929–2936.

[43] Megan M McClelland and Claire E Cameron. “Self-regulation in early childhood: Improving conceptual clarity and developing ecologically valid measures”. In: *Child development perspectives* 6.2 (2012), pp. 136–142.

[44] Brooke SG Molina and William E Pelham Jr. “Childhood predictors of adolescent substance use in a longitudinal study of children with ADHD.” In: *Journal of abnormal psychology* 112.3 (2003), p. 497.

[45] Franziska Mueller et al. “Ganerated hands for real-time 3d hand tracking from monocular rgb”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2018, pp. 49–59.

[46] Meinard Müller. “Chapter 4: Dynamic Time Warping”. In: *Information Retrieval for Music and Motion* (), pp. 69–84. DOI: 10.1007/978-1-4020-6754-9_4969.

[47] Sunita Nayak et al. “Finding recurrent patterns from continuous sign language sentences for automated extraction of signs”. In: *Journal of Machine Learning Research* 13.Sep (2012), pp. 2589–2615.

[48] Carol Neidle, Ashwin Thangali, and Stan Sclaroff. “Challenges in development of the american sign language lexicon video dataset (asllvd) corpus”. In: *Proc. 5th Workshop on the Representation and Processing of Sign Languages: Interactions between Corpus and Lexicon, Language Resources and Evaluation Conference (LREC) 2012*. Citeseer. 2012.

[49] Markus Oberweger and Vincent Lepetit. “Deepprior++: Improving fast and accurate 3d hand pose estimation”. In: *Proceedings of the IEEE International Conference on Computer Vision Workshops*. 2017, pp. 585–594.

[50] Markus Oberweger, Paul Wohlhart, and Vincent Lepetit. “Hands deep in deep learning for hand pose estimation”. In: *arXiv preprint arXiv:1502.06807* (2015).

[51] Himanshu Pahwa. “Handling Periodic Signs in American Sign Language Using Synthetic Generation of Periods”. PhD thesis. University of Texas at Arlington, 2010. ISBN: 9788578110796. DOI: 10.1017/CBO9781107415324.004. arXiv: arXiv:1011.1669v3.

[52] Georgios Pavlakos et al. “Learning to estimate 3D human pose and shape from a single color image”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2018, pp. 459–468.

[53] AJ Piergiovanni and Michael S Ryoo. “Fine-grained activity recognition in baseball videos”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*. 2018, pp. 1740–1748.

[54] Lionel Pigou et al. “Sign language recognition using convolutional neural networks”. In: *Workshop at the European Conference on Computer Vision*. Springer. 2014, pp. 572–578.

[55] Georg Poier, David Schinagl, and Horst Bischof. "Learning pose specific representations by predicting different views". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2018, pp. 60–69.

[56] Georg Poier et al. "MURAUER: Mapping unlabeled real data for label austerity". In: *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*. IEEE. 2019, pp. 1393–1402.

[57] Charles Ruizhongtai Qi et al. "Pointnet++: Deep hierarchical feature learning on point sets in a metric space". In: *Advances in neural information processing systems*. 2017, pp. 5099–5108.

[58] Joseph Redmon and Ali Farhadi. *YOLOv3: An Incremental Improvement*. 2018. arXiv: `1804.02767 [cs.CV]`.

[59] A Ruiz, David H Owens, and Stuart Townley. "Existence, learning, and replication of periodic motions in recurrent neural networks". In: *IEEE Transactions on Neural Networks* 9.4 (1998), pp. 651–661.

[60] Abigail Emma Russell, Tamsin Ford, and Ginny Russell. "Socioeconomic associations with ADHD: findings from a mediation analysis". In: *PloS one* 10.6 (2015), e0128248.

[61] Marin Šaric. "Libhand: A library for hand articulation". In: *Version 0.9* (2011).

[62] Karen Simonyan and Andrew Zisserman. "Very deep convolutional networks for large-scale image recognition". In: *arXiv preprint arXiv:1409.1556* (2014).

[63] Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. "UCF101: A dataset of 101 human actions classes from videos in the wild". In: *arXiv preprint arXiv:1212.0402* (2012).

[64] Adrian Spurr et al. "Cross-modal deep variational hand pose estimation". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2018, pp. 89–98.

[65] Thad Starner and Alex Pentland. “Real-time american sign language recognition from video using hidden markov models”. In: *Motion-Based Recognition*. Springer, 1997, pp. 227–243.

[66] Arlene R Stiffman et al. “A brief measure of children’s behavior problems: The Behavior Rating Index for Children”. In: *Measurement and Evaluation in Counseling and Development* 17.2 (1984), pp. 83–90.

[67] Xiao Sun et al. “Cascaded hand pose regression”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015, pp. 824–832.

[68] James S Supancic et al. “Depth-based hand pose estimation: data, methods, and challenges”. In: *Proceedings of the IEEE international conference on computer vision*. 2015, pp. 1868–1876.

[69] Muhammed Miraç Süzgün et al. “HospiSign: an interactive sign language platform for hearing impaired”. In: *Deniz Bilimleri ve Mühendisliği Dergisi* 11.3 (2015).

[70] Eugene M. Taranta II et al. “A Rapid Prototyping Approach to Synthetic Data Generation for Improved 2D Gesture Recognition”. In: *Proceedings of the 29th Annual Symposium on User Interface Software and Technology*. UIST ’16. Tokyo, Japan: ACM, 2016, pp. 873–885. ISBN: 978-1-4503-4189-9. DOI: 10.1145/2984511.2984525. URL: http://doi.acm.org/10.1145/2984511.2984525.

[71] Stavros Theodorakis, Vassilis Pitsikalis, and Petros Maragos. “Dynamic–static unsupervised sequentiality, statistical subunits and lexicon for sign language recognition”. In: *Image and Vision Computing* 32.8 (2014), pp. 533–549.

[72] Beth L Tieman, Robert J Palisano, and Ann C Sutlive. “Assessment of motor development and function in preschool children”. In: *Mental retardation and developmental disabilities research reviews* 11.3 (2005), pp. 189–196.

[73] Jonathan Tompson et al. “Real-time continuous pose recovery of human hands using convolutional networks”. In: *ACM Transactions on Graphics (ToG)* 33.5 (2014), pp. 1–10.

[74] Clayton Valli. *The Gallaudet Dictionary of American Sign Language*. Gallaudet University Press, 2005. ISBN: 1563682826.

[75] Duncan P Van Dusen et al. "Associations of physical fitness and academic performance among schoolchildren". In: *Journal of School Health* 81.12 (2011), pp. 733–740.

[76] Ulrich Von Agris et al. "Rapid signer adaptation for isolated sign language recognition". In: *2006 Conference on Computer Vision and Pattern Recognition Workshop (CVPRW'06)*. IEEE. 2006, pp. 159–159.

[77] Chengde Wan et al. "Crossing nets: Combining gans and vaes with a shared latent space for hand pose estimation". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2017, pp. 680–689.

[78] Chengde Wan et al. "Self-supervised 3d hand pose estimation through training by fitting". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2019, pp. 10853–10862.

[79] Haijing Wang et al. "A system for large vocabulary sign search". In: *European Conference on Computer Vision*. Springer. 2010, pp. 342–353.

[80] Sy Bor Wang et al. "Hidden Conditional Random Fields for Gesture Recognition". In: *Proceedings of IEEE Computer Vision and Pattern Recognition* (), pp. 1521–1527. ISSN: 1063-6919. DOI: 10.1109/CVPR.2006.132.

[81] Erik G Willcutt et al. "Validity of the executive function theory of attention-deficit/hyperactivity disorder: a meta-analytic review". In: *Biological psychiatry* 57.11 (2005), pp. 1336–1346.

[82] Shanxin Yuan et al. "Bighand2. 2m benchmark: Hand pose dataset and state of the art analysis". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2017, pp. 4866–4874.

[83] Shanxin Yuan et al. “Depth-based 3d hand pose estimation: From current achievements to future goals”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2018, pp. 2636–2645.

[84] Philip David Zelazo et al. “II. NIH Toolbox Cognition Battery (CB): Measuring executive function and attention”. In: *Monographs of the Society for Research in Child Development* 78.4 (2013), pp. 16–33.

[85] Jing Zhang et al. “RGB-D-based Action Recognition Datasets: A Survey”. In: Wanqing Li (2016). arXiv: `1601.05511`. URL: `http://arxiv.org/abs/1601.05511`.

[86] Christian Zimmermann and Thomas Brox. “Learning to estimate 3d hand pose from single rgb images”. In: *Proceedings of the IEEE international conference on computer vision*. 2017, pp. 4903–4911.