

Fine-Grained Activity Detection in the Kitchen with UWB

University of Alberta



Steven Phan

08 April 2023

Contents

1	Introduction	4
2	Literature Review	6
2.1	Overview	7
2.2	Background for Diseases and Conditions in Aging	9
2.2.1	Frailty	9
2.2.2	Dementia	10
2.2.3	Hearing Loss	10
2.2.4	Vision Loss	10
2.2.5	Dizziness	10
2.2.6	Heart Failure	11
2.2.7	Ischemic Heart Disease	11
2.2.8	Diabetes Mellitus	12
2.2.9	Osteoarthritis	12
2.2.10	Osteoporosis	13
2.3	Traditional Assessment of ADLs	13
2.3.1	Barthel index	15
2.3.2	Katz Index of Independence in ADLs	16
2.3.3	Functional Independence Measure	17
2.3.4	ADL Profile	18
2.3.5	ADL Questionnaire	19
2.3.6	Australian Therapy Outcome Measures	21
2.3.7	Melbourn Low-Vision ADL Index	22
2.3.8	Self-Assessment PD Disability Scale	22
2.3.9	Frenchay Activities Index	25
2.3.10	ADL Profile Instrumental	26
2.3.11	Lawton Instrumental ADL Scale	27
2.3.12	Performance Assessment of Self-Care Skills	29

2.3.13	Texas Functional Living Scale	29
2.3.14	Summary	29
2.4	In-Home Monitoring of ADLs	32
2.4.1	Camp et al's Tech Used to Recognize ADL in Community-Dwelling Older Adults	32
2.4.2	Gadey et al's Tech for Monitoring ADL in Older Adults	37
2.5	Motivation	39
2.6	Monitoring Cooking	39
2.7	Next Steps	43
3	System Testing and Tuning at the Independent Living Suite	45
3.1	System Tuning Review	46
3.2	Methodology	46
3.2.1	Setup	46
3.2.2	Protocol	49
3.3	Results	50
3.4	Discussion	54
3.4.1	X Position	54
3.4.2	Y Position	54
3.4.3	Z Position	54
3.4.4	Overall	55
4	Classification of Time Series Data	57
4.1	Classic Machine Learning Methods	58
4.1.1	k-Nearest Neighbours (kNN)	58
4.1.2	Support Vector Machines (SVM)	60
4.1.3	Random Forests	67
4.1.4	Shapelet Transform	70
4.2	Neural Networks	73
4.2.1	Overview	73
4.2.2	Deep Neural Networks (DNN)	74
4.2.3	Convolutional Neural Networks (CNN)	81
4.2.4	Recurrent Neural Networks (RNN)	87
4.2.5	Long Short Term Memory (LSTM) Neural Networks .	90
4.2.6	Transformer Neural Networks	93
4.3	Classification Strategy	99

5 Pilot Testing of Detecting Activities to make a Sandwich	101
5.1 Experimental Protocol	104
5.1.1 Setup	104
5.1.2 Data Collection	105
5.2 Feature Extraction	106
5.3 Model Selection	108
5.4 Results	108
5.5 Discussion	115
6 PASS Tasks	116

Chapter 1

Introduction

This thesis project investigates how to detect fine-grained action within the meal preparation activity of daily living (ADL) in the home without the use of privacy-intruding cameras. ADLs are common activities that an individual performs inside their homes. These include walking around, eating, dressing, personal hygiene, toileting, transportation, meal preparation, house cleaning, and managing medication. The meal preparation ADL was chosen as the main focus because cooking is a uniquely enjoyable activity while being procedurally dense. Meal preparation can include the following actions: opening the fridge, retrieving ingredients, cutting vegetables, and assembling the ingredients. Monitoring these actions may be used as part of a health monitoring program by enabling the assessment of the presence, duration, and correctness of each individual step in a goal-orientated activity. Missing or incorrect steps can be indicative of forgetfulness, and steps that take a long time can be indicative of low-efficacy or struggle that clinicians can address. Information obtained through the monitoring of the cooking task may help guide interventions and track the effectiveness of interventions in clinical populations such as people with dementia, and frailty. A further and more in-depth review of the background will be done in Chapter 2.

It is hypothesized that the combination of context, such as accurate indoor localization (down to 30 cm), and inertial data can accurate and reliable classification of fine-grained ADLs. The system used for indoor localization is the Pozyx Creator Kit which provides a wrist mounted wearable that can obtain data at a maximum of 60 Hz [1]. This data includes position relative to a floorplan, and inertial data from a BNO055 which outputs 3D Acceleration, 3D angular velocity, 3D Linear Acceleration, and the Heading, Pitch and Roll.

Prior to any experiments related to classification of these cooking actions, the optimal configuration of the system that provides reliable position data had to be investigated. Chapter 3 details the different attempts at changing the configuration of the Pozyx Creator Kit to obtain the most reliable positioning data in X, Y, and Z at a satisfactory sampling rate.

Chapter 2

Literature Review

2.1 Overview

In 2014, over 6 million Canadians (15.6% of the population) were 65 years old or older. The number of older adults (65+) continues to increase; and by 2030, Statistics Canada expects there to be over 9.5 million adults over the age of 65, comprising 23% of Canadians [2]. Frequently, older adults who wish to age-in-place or in the comfort of their own home must be able to perform their activities of daily living (ADLs) while bearing multiple diseases and syndromes that come with age, such as frailty, impaired cognition, gait and balance problems [3]. These ADLs may include cooking, bathing, getting into and out of bed, and toileting all of which require complex coordination of the older adult's cognitive, physical, visual, and perceptual abilities (a complete list may be found in Table 2.1). Deficits in any of the categories mentioned can impair the older adult's ability to go about their day. However, it is typically only after an incident or hospitalization that an older adult is assessed for their ability to perform ADLs [4].

The current system presents an opportunity for proactive and preventative medicine through the use of in-home monitoring. Data obtained through monitoring can be used to track functional decline. With this information, older adults, along with their clinician, can plan early interventions and prevent future incidents and hospitalization.

Table 2.1: Complete list of Basic ADLs and Instrumental ADLs. Adapted from [5].

Category	Name	Description
Basic ADLs	Ambulating	Ability to move from one position to another and get around the house.
	Feeding	The ability to feed oneself.
	Personal Hygiene	The ability to maintain hygiene for oneself. Includes dental, nails and hair hygiene.
	Continence	The ability to put on clothes, shoes, pants, etc.
	Toileting	The ability to use the toilet appropriately and clean oneself after.
Instrumental ADLs	Transportation and Shopping	Ability to arrange transportation for oneself or drive. Also, the ability to procure groceries, clothing and any other daily necessities.
	Managing Finances	Ability to manage assets and pay the bills.
	Meal preparation	End-to-end ability to put a meal onto the table. Involves both purchasing of groceries, cooking, plating and bringing to the dining table.
	Housecleaning and home maintenance	Cleaning kitchen after eating and keeping living areas reasonably clean and tidy. The ability to maintain the home and arrange repairs if needed.
	Managing communication with other	The ability to use phone and mail to communicate with others.
	Managing Medication	Ability to obtain medications and take them as directed.

To better understand the role of in-home monitoring, the diseases and conditions that older adults may face as they age must be understood. Accordingly, the next sections will be organized as follows: first, an overview of the disease or condition including its definition, prevalence and symptoms will be presented; next, current assessment tools for ADLs will be discussed; finally, current attempts at in-home monitoring for ADLs will be investigated as well as what data and how the data is being used.

2.2 Background for Diseases and Conditions in Aging

This section reviews the current literature on the disease and conditions associated that affect the ability to perform ADLs as older adults age. In terms of functionally being able to perform one's ADLs, both physical and cognitive ability are required. Peter et al., in their exploration of ADLs, mention frailty and dementia as conditions limiting ADL performance [5]. In addition to physical and cognitive abilities, sensory decline with hearing, vision and vestibular function (associated with dizziness) occur as one ages [6]. Several chronic conditions may also be present in older adults such as cardiovascular disease including heart failure and ischemic heart disease; diabetes mellitus; osteoarthritis; and osteoporosis [6].

2.2.1 Frailty

Frailty is a syndrome present in 20-50% of the middle and older aged population (ages 50+ years) [7] and is associated with ageing and co-morbidities, but not caused by them [8]. Individuals with frailty are at “higher risk for adverse health outcomes such as illnesses, hospitalization, disability and mortality”. To define frailty, there are 2 models: the frailty phenotype and the frailty index [9]. The frailty phenotype (also known as the Fried’s Definition of Cardiovascular Health Study) defines frailty as meeting three out of five of the following criteria: “weakness, slowness, low level of physical activity, self-reported exhaustion and unintentional weight loss,” whereas the frailty index uses a comprehensive geriatric assessment to determine cumulative deficits [9].

2.2.2 Dementia

Worldwide, 47 million people live with dementia, and the number is only expected to increase. By 2050, it is projected that 131 million individuals will be living with dementia [10]. In the US, the prevalence of dementia in older adults above the age of 68 is 15% [10]. Dementia is an umbrella term describing a gradual decline in cognitive abilities in several domains to the point of impairing social or occupational function [10]. It involves a slow onset and gradual loss of memory, typically paired with the inability to retain new information and ability to perform daily activities [10]. Prior to diagnosing dementia, there is typically a long history of cognitive decline combined with mention of cognitive decline from close friends and family [10]. When diagnosing dementia, a common tool that is used is the Mini Mental State Examination (MMSE): a paper test that lists several simple tasks such as spelling a word backwards, and asking about the day of the week, etc. [11].

2.2.3 Hearing Loss

The prevalence of hearing loss increases with age and at around 85 years or older, about half of all older adults experience hearing impairments [12]. Hearing loss is associated with a decreased quality of life and impairs speech processing. Reduced conversational ability may lead to social isolation which is associated with depression and cognitive decline [6].

2.2.4 Vision Loss

Similar to hearing loss, the prevalence of vision loss increases with age. In the UK, it was found that 23% of older adults ages 85-89 had severe vision loss and this increases to 37% for older adults older than 90 years [13]. Older adults with vision loss are reported to have slower walking speeds and have difficulty doing physical activities [14]. Vision loss can also "predict cognitive decline and risk of dementia" [14].

2.2.5 Dizziness

In 2000, Tinetti et al., found that 24% of their sample American population over age 72 experience dizziness [15]. The cause of dizziness in the older pop-

ulation is multifactorial, but the most common cause is peripheral vestibular dysfunction (the part of the ear responsible for balance) [16]. As one ages, the number of vestibular hair cells and neurons decrease which impairs the older adult's ability to sense changes in orientation [16]. As a result, their balance and postural stability deteriorates and they are at higher risk of falling [16].

2.2.6 Heart Failure

Heart Failure affects at least 26 million people worldwide and the number of people affected continues to grow [17]. The American College of Cardiology (ACC) Foundation and American Heart Association (AHA) define Heart failure (HF) as “a complex clinical syndrome that results from any structural or functional impairment of ventricular filling or ejection of blood [18].” There are two types of HF: HF with reduced ejection fraction and HF with preserved ejection fraction. HF with reduced ejection fraction occurs at an ejection fraction $\leq 40\%$ and HF with preserved ejection fraction occurs at an ejection fraction $\geq 50\%$ [18]. Though patients with HF present with a variety symptoms, some include shortness of breath, lethargy, fatigue, reduced exercise tolerance, wheezing, and ankle swelling [19].

2.2.7 Ischemic Heart Disease

Ischemic heart disease (IHD) is also called coronary heart disease (CHD) or coronary artery disease (CAD) [20] and between the periods of 2003 to 2006 it was estimated that 17.6 million or 7.9% of Americans age 20 or older had CAD [20]. Similar to the other conditions, prevalence of IHD increases with age: in 2017 the National Health and Nutrition Examination Survey reported a prevalence of 30.6% in men and 21.7% in women over the age of 80 [21]. IHD manifests due to a condition called atherosclerosis which is a buildup of plaque in the blood vessels [20]. Atherosclerosis reduces the diameter of the blood vessel which reduces the supply of blood to an organ. The condition when an organ doesn't get enough blood and oxygen is referred to as ischemia. In the case of Ischemic Heart Disease, the heart does not receive enough blood and oxygen [20]. As a result, IHD may cause symptoms such as fatigue and angina pectoris which is the discomfort that is felt when the heart muscle does not get enough oxygen [20].

2.2.8 Diabetes Mellitus

Diabetes occurs when the body cannot control its blood sugar levels [22]. People who have diabetes are at risk of damaging blood vessels in their eyes, kidneys and nerves [22]. Diabetes mellitus can come in two main subtypes: type 1 diabetes mellitus (T1DM) or type 2 diabetes mellitus (T2DM) [22]. T1DM occurs due to the destruction of insulin producing beta-cells from an autoimmune process, while T2DM appears when cells develop a resistance to insulin and fail to use insulin that is being produced. The full extent of T2DM occurs when a cell's resistance to insulin overtakes the body's ability to produce insulin [22]. Diabetes affects 1 in 11 adults. 90% of adults have T2DM and the remaining 10% have T1DM. Onset for people with T1DM gradually increases from birth and peaks at ages 4 to 6 years and again from 10 to 14 years [22]. Signs and symptoms of diabetes include being overweight/obese, blurry vision, yeast infections, numbness, and neuropathic pain [22]. T1DM is diagnosed based on a history of "fasting glucose over 126 mg/dL, random glucose over 200 mg/dL, or hemoglobin A1C exceeding (HbA1C) 6.5%" [22]. The early stages of T2DM uses fasting glucose and HbA1C as well, but Sapra and Bhandari do not provide a specific level. A precursor to T2DM, prediabetes can be diagnosed with a glucose level of 100 to 125 mg/dL or a 2-hour post-oral glucose tolerance test (testing blood sugar levels one hour after ingesting 75 g of glucose dissolved into 250 ml to 300 ml of water [23]) glucose level of 140 to 200 mg/dL [22].

2.2.9 Osteoarthritis

Osteoarthritis affects approximately 27 million Americans [24] and onsets in a third of adults during the typical working age of 45-64 years [25]. There are about 315 million visits to the doctor, and 744 000 hospital admissions per year in the US for osteoarthritis. These figures add up to a total of 68 million days off work [26]. Osteoarthritis is the most common type of joint disorder. It is a disease characterized by the mechanical destruction and failure of a synovial joint [27]. In order of the most common location, the knee ranks first, followed by the hand and the hip [27]. High-risk factors or factors that greatly increase the chance of having osteoarthritis include obesity and previous joint injury [26]. Osteoarthritis typically presents as debilitating pain, and is accompanied by stiffness, reduced range of motion, joint instability, swelling, muscle weakness, and fatigue [27]. Clinical diagnosis is based on the

symptoms that the patient presents with such as pain, and functional limitations [27]. Additionally, diagnostic criteria such as those from the American College of Rheumatology may also be used [27, 28].

2.2.10 Osteoporosis

Osteoporosis is characterized by a reduction in bone mass and structural deterioration inside of the bone [29]. It is estimated that over 200 million people have osteoporosis and incidence increases with age [30]. Over the age of 80, it is estimated that 70% of people have osteoporosis [30]. This condition leads to an increased risk of bone fractures at all bone locations, with hip and vertebral fractures being historically associated with osteoporosis [29]. Typically, there is a long "latent period," where osteoporosis goes undetected, before it manifests clinically as a vertebral, rib or hip fracture [31]. One of the earliest symptoms of osteoporosis occurs as a result vertebral compression fractures [31]. The individual may experience acute back pain at rest or during activity such as bending, standing from a seated position and lifting an object [31]. Individuals with hip fractures experience pain and an inability to bear weight which leads to reduced functional status and quality of life [29]. In 2010, 2.7 million hip fractures occurred worldwide [29] and as one ages, the incidence of hip fractures increases exponentially [29].

2.3 Traditional Assessment of ADLs

The traditional assessment of ADLs often involves a pen and paper test. Some tools may require a trained clinician to ask the patient to perform some tasks related to the ADL and observe their performance [32]. Based on the criteria outlined in the assessment tool, a score is assigned and helps the clinician later identify deficits and plan care accordingly [33]. Other tools may require the clinician to check patient medical records, use direct observation or interview the patient directly to assign scores [34].

Pashmdarfard and Azad in their systematic review of ADL assessment tools identified 8 tools assessing Basic ADLs and 5 tools assessing Instrumental ADLs [32]. The Basic ADLs assessment tools are as follows:

- Barthel index
- Katz Index of Independence in ADLs

- Functional Independence Measure
- ADL Profile
- ADL Questionnaire
- Australian Therapy Outcome Measures
- Melbourn Low-Vision ADL Index
- Self-Assessment Parkinson's disease (PD) Disability Scale

And the Instrumental ADL assessment tools are as follows:

- Frenchay Activities Index
- ADL Profile Instrumental
- Lawton Instrumental ADL Scale
- Performance Assessment of Self-Care Skills
- Texas Functional Living Scale

2.3.1 Barthel index

There are 10 activities related to Basic ADLs that are assessed by the Barthel Index: bowels, bladder, grooming, toilet use, feeding, transfer, mobility, dressing, stairs, and bathing [35]. Total possible scores range from 0-20 with 20 being highly functional and 0 indicating high disability. Scores can be obtained through self-report, by proxy (reports from someone who is familiar with the individual), or through observation. A version of the Barthel Index used at Alberta Health Services is provided in Figure 2.1.



Barthel Index of Activities of Daily Living

Instructions: Choose the scoring point for the statement that most closely corresponds to the patient's current level of ability for each of the following 10 items. Record actual, not potential, functioning. Information can be obtained from the patient's self-report, from a separate party who is familiar with the patient's abilities (such as a relative), or from observation. Refer to the Guidelines section on the following page for detailed information on interpretation.

The Barthel Index

Bowels 0 = incontinent (or needs to be given enemas) 1 = occasional accident (once/week) 2 = continent <i>Patient's Score:</i> _____	Transfer 0 = unable – no sitting balance 1 = major help (one or two people, physical), can sit 2 = minor help (verbal or physical) 3 = independent <i>Patient's Score:</i> _____
Bladder 0 = incontinent, or catheterized and unable to manage 1 = occasional accident (max once per 24 hours) 2 = continent (for over 7 days) <i>Patient's Score:</i> _____	Mobility 0 = immobile 1 = wheelchair independent, including corners, etc. 2 = walks with help of one person (verbal or physical) 3 = independent (but may use any aid, e.g., stick) <i>Patient's Score:</i> _____
Grooming 0 = needs help with personal care 1 = independent face/hair/teeth/shaving (implements provided) <i>Patient's Score:</i> _____	Dressing 0 = dependent 1 = needs help, but can do about half unaided 2 = independent (including buttons, zips, laces, etc.) <i>Patient's Score:</i> _____
Toilet Use 0 = dependent 1 = needs some help, but can do something alone 2 = independent (on and off, dressing, wiping) <i>Patient's Score:</i> _____	Stairs 0 = unable 1 = needs help (verbal, physical, carrying aid) 2 = independent up and down <i>Patient's Score:</i> _____
Feeding 0 = unable 1 = needs help cutting, spreading butter, etc. 2 = independent (food provided within reach) <i>Patient's Score:</i> _____	Bathing 0 = dependent 1 = independent (or in shower) <i>Patient's Score:</i> _____

Total Score:_____

Scoring: Sum the patient's scores for each item. Total possible scores range from 0 – 20, with lower scores indicating increased disability. If used to measure improvement after rehabilitation, changes of more than two points in the total score reflect a probable genuine change, and change on one item from fully dependent to independent is also likely to be reliable.

Figure 2.1: Alberta Health Services (AHS) adaptation of the Barthel Index [35]

2.3.2 Katz Index of Independence in ADLs

The Katz Index evaluates 6 categories for Basic ADLs: bathing, dressing, toileting, transfers, continence, and feeding for older adults. Though the Katz Index is binary (either independent or dependent), the evaluator is given 3 choices for each category and must select one for each: independence, intermediate dependence and full dependence [36]. Figure 2.2a shows the evaluation form a clinician would use. Adequacy or overall performance in the 6 functions is given as a grade (A, B, C, D, E, F, and other) based on the type of dependence. The description of these grades may be found in Figure 2.2b. The results of this assessment can be used to describe functional level at the onset of an illness such as hip fracture [36]. It may also be used to compare treatment effectiveness between a control and treatment group, acts as a guide for hospital admittance, as well as guide progress during therapy [36].

Name.....	Date of evaluation.....	
For each area of functioning listed below, check description that applies. (The word "assistance" means supervision, direction or personal assistance.)		
Bathing—either sponge bath, tub bath, or shower.		
<input type="checkbox"/> Receives no assistance (gets in and out of tub or self if tub is usual means of bathing)	<input type="checkbox"/> Receives assistance in bathing only one part of the body (such as back or a leg)	<input type="checkbox"/> Receives assistance in bathing more than one part of the body (or not bathed)
Dressing—gets clothes from closet and drawers—including underclothes, outer garments and using fasteners (including braces if worn)		
<input type="checkbox"/> Gets clothes and gets completely dressed without assistance	<input type="checkbox"/> Gets clothes and gets dressed without assistance except for assistance in tying shoes	<input type="checkbox"/> Receives assistance in getting clothes or stays partly or completely undressed
Toileting—going to the "toilet room" for bowel and urine elimination; cleaning self after elimination, and arranging clothes		
<input type="checkbox"/> Goes to "toilet room," cleans self, and arranges clothes without assistance (may use object for support such as cane, walker, chair, hair chair and may manage night bedpan or commode, emptying same in morning)	<input type="checkbox"/> Receives assistance in going to "toilet room" or in cleaning clothes after elimination or in use of night bedpan or commode	<input type="checkbox"/> Doesn't go to room termed "toilet" for the elimination process
Transfer—		
<input type="checkbox"/> Moves in and out of bed as well as in and out of chair with assistance (may be using object for support such as cane or walker)	<input type="checkbox"/> Moves in or out of bed or chair with assistance	<input type="checkbox"/> Doesn't get out of bed
Continence—		
<input type="checkbox"/> Controls urination and bowel movement completely by self	<input type="checkbox"/> Has occasional "accidents"	<input type="checkbox"/> Supervision helps keep urine or bowel control; catheter is used, or is incontinent
Feeding—		
<input type="checkbox"/> Feeds self without assistance	<input type="checkbox"/> Feeds self except for getting assistance in cutting meat or buttering bread	<input type="checkbox"/> Receives assistance in feeding or is fed partially or completely by tube or intravenous fluids

The Index of Independence in Activities of Daily Living is based on an evaluation of the functional independence or dependence of patients in bathing, dressing, going to toilet, transferring, continence, and feeding. Specific definitions of functional independence and dependence appear below the index.

A — Independent in feeding, continence, transferring, going to toilet, dressing, and bathing.
B — Independent in all but one of these functions.
C — Independent in all but bathing and one additional function.
D — Independent in all but bathing, dressing, and one additional function.
E — Independent in all but bathing, dressing, going to toilet, and one additional function.
F — Independent in all but bathing, dressing, going to toilet, transferring, and one additional function.
G — Dependent in all six functions.
Other — Dependent in at least two functions, but not classifiable as C, D, E, or F.

(a) Evaluation form from the Katz Index of Independence in ADLs [36]

(b) Grades for determining adequacy outlined in [36]

Figure 2.2: The Katz Index of Independence in ADLs [36]

2.3.3 Functional Independence Measure

The Functional Independence Measure (FIM) is frequently used for patients who have experienced stroke [37]. FIM is based off of the Barthel Index and contains 18 items that tests independence in "self-care activities, mobility, locomotion, communication, sphincter control, and cognition" [37]. Each item is rated from 1 (total assistance) to 7 (independent) for a total score that ranges from 18 to 126—a higher score indicates higher independence [37]. If the level of dependence can not be evaluated, then the lowest score of 1 is given [37]. Unfortunately, the FIM is proprietary and cannot be reproduced here [38].

2.3.4 ADL Profile

The ADL Profile was created to assess ADL function in head injured people whose deficits lie mostly with "complex home and community activities" [39]. Accordingly, 10 domains were chosen for assessment: personal hygiene, dressing, feeding, health care, meal preparation and home management, and use of public services, transportation, financial management and time management [39]. Tasks are devised for each domain and the clinician would observe the patient performing the task. Following the completion of a task the clinician would rate the patient on a 3-point ordinal scale with 0 indicating dependence and 2 indicating independence based on their ability to perform the 10 operations common to all of the tasks [39]. See Figure 2.3 for an example of a score sheet for the ADL Profile.

OPERATION		Vigilance, attention learning disposition	Reception, analysis storage of information	Anticipation and goal formulation	Planning	Carrying out and verification	Simple movement	Kinesthesia	Optic-Spatial organization	Dynamic organization	Complex praxis
Y	YES										
N	NO										
-	non applicable										
HYGIENE											
1 completes personal hygiene <input type="checkbox"/> sink <input type="checkbox"/> bath <input type="checkbox"/> shower											
2 performs grooming <input type="checkbox"/> teeth <input type="checkbox"/> underarms <input type="checkbox"/> shave <input type="checkbox"/> nails <input type="checkbox"/> hair											
3 performs toileting (care of perineum, toilet use ...)											
DRESSING											
4 puts on/remove indoor clothing and shoes											
5 puts on/remove outdoor clothing and footwear											
6 puts on/remove aids/devices/ accessories											

Figure 2.3: Example of a score sheet for some tasks in the ADL profile [39]. Columns show the operations for each of the tasks.

Since Dutil et al's 1990 paper on the ADL profile, several other sources seem to have indicated that there have been some updates to the ADL Profile. Notably, the scoring system has expanded to a 4-point ordinal scale with 0 being independent and 3 being completely dependent [32, 40]. Furthermore the assessment includes a questionnaire for 3 items that are assessed through

a semi-structured interview with the person or their caregiver(s) [32, 40].

2.3.5 ADL Questionnaire

The ADL Questionnaire was developed for individuals with Alzheimer's Disease. It is applicable to a wide range of dementia syndromes and can be used to track functional decline over time [41]. It assess six areas: "self-care, household care, employment and recreation, shopping and money, travel, and communication" [41]. The primary caregiver assigns a score from 0 (no problem) to 3 (unable to perform) for each areas by comparing current level of ability to the level of ability before the onset of dementia" [41]. Impairment is then calculated by formula 2.1 (Note that all scores with "9" are excluded because either the patient has never done the activity or the caregiver doesn't know):

$$\text{Functional Impairment} = \frac{\text{Sum of all ratings}}{3 * \text{total number of items rated}} * 100\% \quad (2.1)$$

Impairment is rated by ranges: 0-33% indicates none to mild impairment, 34-66% indicates moderate impairment and > 66% indicates severe impairment. An excerpt of the questionnaire is provided in Figure 2.4

APPENDIX. Activities of Daily Living Questionnaire (ADLQ)

Instructions: circle one number for each item

1. Self-care activities

A. Eating

- 0 = No problem
- 1 = Independent, but slow or some spills
- 2 = Needs help to cut or pour; spills often
- 3 = Must be fed most foods
- 9 = Don't know

B. Dressing

- 0 = No problem
- 1 = Independent, but slow or clumsy
- 2 = Wrong sequence, forgets items
- 3 = Needs help with dressing
- 9 = Don't know

C. Bathing

- 0 = No problem
- 1 = Bathes self, but needs to be reminded
- 2 = Bathes self with assistance
- 3 = Must be bathed by others
- 9 = Don't know

D. Elimination

- 0 = Goes to the bathroom independently
- 1 = Goes to the bathroom when reminded; some accidents
- 2 = Needs assistance for elimination
- 3 = Has no control over either bowel or bladder
- 9 = Don't know

E. Taking pills or medicine

- 0 = Remembers without help
 - 1 = Remembers if dose is kept in a special place
 - 2 = Needs spoken or written reminders
 - 3 = Must be given medicine by others
 - 9 = Does not take regular pills or medicine **OR** Don't know
-

Figure 2.4: Excerpt of the questions asked in the ADL questionnaire [41]

2.3.6 Australian Therapy Outcome Measures

The Australian Therapy Outcome Measures (AuTOM) is based on the UK Therapy Outcome Measures [42] and is used for evaluating therapy outcome measures across speech pathology, occupational therapy and physiotherapy. A core scale was developed with categories of impairment of structure or function, activity limitations, participation restriction and well-being each rated from 0 (most impairment) to 5 (no impairment). ADLs seem to reside in the occupational therapy domain which have 12 domains [42]:

1. learning and applying knowledge
2. self-care
3. functional walking and mobility
4. domestic life: inside house
5. upper limb use
6. domestic life: outside house
7. carrying out daily life tasks and routines
8. interpersonal interactions and relationships
9. transfers
10. work, employment and education
11. using transport
12. community life, recreation, leisure and play

2.3.7 Melbourn Low-Vision ADL Index

The Melbourn Low-Vision ADL Index was created to address the need to assess the ADL performance of individuals with vision impairment [43]. The index contains both task-based observational rating as well as questionnaire self-reported rating. For both ratings a scale of 0 (unsatisfactory) to 4 (satisfactory) was used. For the task-based observational rating, standardized descriptions at each score were developed to reduce the variability. Performance was evaluated based on speed, accuracy and independence and it was suggested that the evaluator time the task to better assess speed of execution [43]. The list of tasks is shown in Table 2.2.

2.3.8 Self-Assessment PD Disability Scale

The Self-Assessment PD Disability Scale is a self-report scale that bases its assessment on the degree of PD's interference on the individual's daily life [44]—and not on the "amplitude of tremors, or speed of foot tapping" [44]. Accordingly, the individual themselves should know PD's degree of interference on their daily life. There are a total of 24 items on the questionnaire with 11 assessing gross mobility and 13 assessing fine coordination. The individual is asked to rate themselves on a 5-point scale with 1 indicating no interference and 5 indicating maximum interference from PD [44]. The total score ranges from 24 to 120 with higher scores indicating higher disability [44]. The full list of items can be found in Table 2.3.

Table 2.2: Observational and questionnaire items in the Melbourn Low-Vision ADL Index [43]

Observational	Questionnaire
Reading newspaper print	Eating
Reading newspaper headlines	Bathing
Reading a letter with typed print	Dressing
Using a telephone book	Grooming
Reading an account	Mobility
Reading a medicine label	Housework
Reading packet labels	Shopping
Recognizing faces	Preparing meals
Using a telephone	Managing medication
Writing a check	
Identifying coins	
Pouring	
Naming colors	
Buttoning a shirt	
Threading a sewing needle	
Telling the time: wrist watch	
Telling the time: wall clock	
Reading a digital display	

Table 2.3: Items for the Self-Assessment PD Disability Scale [44]

Gross Mobility	Fine Coordination
Getting out of bed	Brushing teeth
Getting out of chair	Washing
Walking around home	Opening tins
Walking outside, eg. to shops	Pouring milk from bottle
Traveling by public transport	Making cup of tea
Walking up stairs	Holding cup and saucer
Walking downstairs	Washing and drying dishes
Getting into bath	Using knife and fork
Getting out of bath	Inserting electrical plug
Getting undressed	Dialing telephone
Picking up object from floor	Holding and reading newspaper Writing letter Getting dressed

2.3.9 Frenchay Activities Index

The Frenchay Activities Index (FAI) was created with 2 goals: first to "provide accurate information for pre-morbid lifestyle for individuals with stroke" and second "record changes in activities following stroke, at specific intervals" [45]. Through factor analysis, 3 factors were identified: "domestic chores", "leisure/work" and "outdoor activities" [45]. The assessment is questionnaire-based and includes 15 items that represent these 3 factors. Each item is given a score of 1 to 4 with 4 indicating more activity. The questionnaire is show in Figure 2.5.

<i>Activities Index</i>	
<i>During previous 3 months</i>	
<i>Activity</i>	<i>Code</i>
Preparing main meals	1 = Never
Washing-up	2 = Under once weekly 3 = 1-2 times a week 4 = Most days
Washing clothes	1 = Never
Light housework	2 = 1-2 times in 3 months
Heavy housework	3 = 3-12 times in 3 months
Local shopping	4 = At least weekly
Social outings	
Walking outdoors over 15 min	
Pursuing active interest in hobby	
Driving a car/travel on bus	
<i>During previous 6 months</i>	
Outings/car rides	1 = Never 2 = 1-2 times in 6 months 3 = 3-12 times in 6 months 4 = At least weekly
Gardening	1 = None
Household and/or car maintenance	2 = Light 3 = Moderate 4 = All necessary
Reading books	1 = None 2 = 1 in 6 months 3 = Less than 1 a fortnight 4 = Over 1 a fortnight
Gainful work	1 = None 2 = Up to 10 h/week 3 = 10-30 h/week 4 = Over 30 h/week
Total ____; Factor 1 ____; Factor 2 ____; Factor 3 ____	

Figure 2.5: The questionnaire used in the Frenchay Activities Index.

2.3.10 ADL Profile Instrumental

The ADL Profile Instrumental v2 is an updated version of the ADL Profile designed specifically to assess Instrumental ADLs in patients with Traumatic Brain Injury [40]. The scoring system uses a 5-level ordinal scale to finely rate independence [40]. Rather than a separate task for each category of evaluation, a single scenario was given to the patient to assess a total of 8 categories: 6 related to the goal of preparing a meal including "dressing to go outdoors, going to the grocery store, shopping for food, preparing a hot meal for guests, having a meal with guests, and cleaning up after the meal" and 2 related to complex single tasks including "obtaining information, making a budget" [40]. In total, there are 29 scores for 5 tasks with 4 operations and 3 tasks with 3 operations [40]. The scenario prompt involved the following script followed by observation of the patient performing the task:

"We would like to know how you manage in your everyday activities, that is, activities that you generally do inside and outside of your home. More specifically, we would like to know, following your accident, if any changes have occurred in your ability to carry out your everyday tasks. Without knowing it, you invited my assistant and me to have lunch with you. Please get ready to receive us. We will assume any incurred expenses for a maximum of \$20. Can you tell me in your own words what I have just explained to you? Would you agree to do this? Can you now tell me in your own words what you are going to do?" [40]

2.3.11 Lawton Instrumental ADL Scale

The Lawton Instrumental ADL Scale was made for older adults and assesses 8 items: telephoning, shopping, food preparation, housekeeping, laundering, use of transportation, use of medicine, and financial behavior [46]. Unlike some of the other scales, there is a distinction in the scoring system between males and females. Males are rated on a 5-point scale whereas females are rated on a 8-point scale. Each item assessed has score of 1 for being able to or somewhat able to perform the task (see Figure 2.6), a score of 0 would indicate that the older adult is unable to perform the task. Women are assessed on all 8 items whereas men are assess on a subset of the items since the "list of representative activities is is smaller" [46]. The list for men include telephoning, shopping, use of transportation, use of medicine, and financial behavior. The Lawton IADL Scale provides a snapshot of what services or treatments the older adult needs (or does not need) and can be re-applied at periodic intervals to update treatment goals [46].

	Male			Female			Male			Female		
	Score	% Correct	% Error	Score	% Correct	% Error	Score	% Correct	% Error	Score	% Correct	% Error
	64	5.2	A. Ability to use telephone	68	4.8					E. Laundry	41	6.0
1			1. Operates telephone on own initiative—looks up and dials numbers, etc.	1					1. Does personal laundry completely.	1		
1			2. Dials a few well-known numbers.	1					2. Launder small items—rinses socks, stockings, etc.	1		
1			3. Answers telephone but does not dial.	1					3. All laundry must be done by others.	0		
0			4. Does not use telephone at all.	0			27	4.1	F. Mode of Transportation	30	10.0	
15	5.2	B. Shopping		15	3.0		1		1. Travels independently on public transportation or drives own car.	1		
1		1. Takes care of all shopping needs independently.	1				1		2. Arranges own travel via taxi, but does not otherwise use public transportation.	1		
0		2. Shops independently for small purchases.	0				0		3. Travels on public transportation when assisted or accompanied by another.	1		
0		3. Needs to be accompanied on any shopping trip.	0				0		4. Travel limited to taxi or automobile with assistance of another.	0		
0		4. Completely unable to shop.	0				35	4.1	G. Responsibility for own Medications	38	9.5	
		C. Food Preparation		20	2.4		0		1. Is responsible for taking medication in correct dosages at correct time.	1		
		1. Plans, prepares and serves adequate meals independently.	1				0		2. Takes responsibility if medication is prepared in advance in separate dosages.	0		
		2. Prepares adequate meals if supplied with ingredients.	0				35	4.1	3. Is not capable of dispensing own medication.	0		
		3. Heats and serves prepared meals, or prepares meals but does not maintain adequate diet.	0				0		5. Does not travel at all.	0		
		4. Needs to have meals prepared and served.	0									
		D. Housekeeping		51	7.1		0					
		1. Maintains house alone or with occasional assistance (e.g., "heavy work-domestic help").	1				54	5.2	H. Ability to Handle Finances	52	10.0	
		2. Performs light daily tasks such as dish-washing, bedmaking.	1				1		1. Manages financial matters independently (budgets, writes checks, pays rent, bills, goes to bank), collects and keeps track of income.	1		
		3. Performs light daily tasks but cannot maintain acceptable level of cleanliness.	1				1		2. Manages day-to-day purchases, but needs help with banking, major purchases, etc.	1		
		4. Needs help with all home maintenance tasks.	1				1		3. Incapable of handling money.	0		
		5. Does not participate in any housekeeping tasks.	0				0					
Rep. = .96	N = 97		Rep. = .93	N = 168	Rep. = .96	N = 97		Rep. = .93	N = 168			

Figure 2.6: The Lawton Instrumental ADL Scale, the blank "score" column shows the items that males are not assessed on [46].

2.3.12 Performance Assessment of Self-Care Skills

The Performance Assessment of Self-Care Skills (PASS) is a performance-based observational tool in which the clinician asks the patient to perform the required task for each of the ADLs assessed and evaluates their performance [32]. PASS assesses an individual's ability to perform ADLs by judging 3 parameters: independence, safety and adequacy. There are concrete guidelines and identifiers mentioned for scoring each parameter in Performance Assessment of Self-Care Skills [47]. For instance, the safety category has a maximum score of 3. At a score of 3, there are no risks observed; at a score of 2, there are minor risks observed, but no assistance is needed; at a score of 1, there are obvious risks to safety and assistance is required to complete a task; Finally, at a score of 0, there are risks to safety of such severity that the task had to be stopped. PASS is used around the world; has been translated to multiple languages including Spanish, Hebrew and Mandarin; has a test-retest reliability of 89%-90%; and an inter-observer agreement of 96%-97% [48] making it a reliable tool for assessing ADLs. However, as much as the PASS is reliable and comprehensive, it is also time-consuming and may be strenuous for the older adults [32].

2.3.13 Texas Functional Living Scale

The Texas Functional Living Scale is a performance-based assessment tool that was created to evaluate the IADL ability of individuals with cognitive impairment. 15 to 20 minutes are required to evaluate 21 items that target 5 functional areas: Dressing, Time, Money, Communication and Memory [49]. The minimum score for each item is 0, and the maximum score varies depending on the item. Altogether, the maximum score attainable is 52, with "higher scores indicating better performance" [49]. Examples of items on the Texas Functional Living Scale is shown in Figure 2.7.

2.3.14 Summary

Several key assessment tools for BADLs and IADLs have been discussed. Table 2.4 summarizes the category of ADL it assesses, what condition it is used for primarily, method of assessment as well as the length of assessment.

Domain	Points	Task
Dressing	5	1. Puts on jacket 2. Ties shoelaces
Time	15	1. States time on clock 2. Calculates time interval on clock 3. Sets clock 4. Locates current date on calendar 5. Reads calendar
Money	12	1. Counts money 2. Pays specified amount 3. Makes change 4. Writes sample check
Communication	12	1. Addresses envelope 2. Calls home 3. Looks up designated telephone number 4. Knows emergency telephone number 5. Describes how to make peanut butter and jelly sandwich
Memory	8	1. Takes out three pills from bottle when timer sounds 2. Recalls payee of check 3. Recalls amount of check
Total	52	

Figure 2.7: An example of some items on The Texas Functional Living Scale.

Table 2.4: Summary of the ADL Assessment tools discussed.

Name	Type	Condition	Method	Length
Barthel Index	BADL	N/A	Observation/Self Questionnaire	2-5 min [50]
Katz Index of Independence in ADLs	BADL	Older Adults or Chronically Ill	Observation Questionnaire	<5 min [51]
Functional Independence Measure	BADL	Stroke	Observation Questionnaire	30 min [52]
ADL Profile	BADL	Head Injured	Performance	30-60 min [53]
ADL Questionnaire	BADL	Alzheimer	Questionnaire	5-10 min [41]
Australian Therapy Outcome Measures	BADL	N/A	Observation Questionnaire	<5 min [54]
Melbourne Low-Vision ADL Index	BADL	Visually Impaired	Performance	20 min [43]
Self-Assessment PD Disability Scale	BADL	Parkinson's	Self Questionnaire	5 min [55]
Frenchay Activities Index	IADL	Stroke	Questionnaire	5 min [56]
ADL Profile Instrumental	IADL	Head Injured	Performance	3 hours [40]
Lawton Instrumental ADL Scale	IADL	Older Adults	Self Questionnaire	10-15 min [57]
Performance of Self-Care Skills	IADL & BADL	Older Adults	Performance	3 hours [58]
Texas Functional Living Scale	IADL	Cognitive Impairment	Performance	15-20 min [59]

2.4 In-Home Monitoring of ADLs

Although traditional pen-and-paper assessment of ADLs can provide information regarding the current ability of an individual to perform ADLs, all only provide snapshots of their current ability and further tracking requires the patient to be present for the assessment. Furthermore, most involve an evaluator training period to ensure evaluator reliability, ratings are subjective, and performance-based evaluations are lengthy [58, 52, 40, 59]. There are also issues with measurement of functional performance in a clinical setting as opposed to an at-home setting that may skew observations. For example, it may be easier to stand-up from a firm higher chair in clinical settings than a soft sofa found at home [60]. In light of these challenges, this section will explore recent advances in monitoring technology for the evaluation of ADL ability. There will be a focus on the type of technology used, the parameters or metrics evaluated, the ADL it is used for and the criteria for assessing function. Information will be drawn primarily from two systematic reviews: Camp et al's "Technology Used to Recognize Activities of Daily Living in Community-Dwelling Older Adults" [60] and Gadey et al.'s "Technologies for monitoring activities of daily living in older adults: a systematic review" [61].

2.4.1 Camp et al's Tech Used to Recognize ADL in Community-Dwelling Older Adults

Camp et al.'s search identified a total of "21 [distinct] technologies, including six wearable sensors, 13 environmental sensors, and two types of camera" [60]. The full list can be found in Table 2.6. These technologies were able to detect a total of 14 different ADLs, but no single technology system was able to detect all 14 [60]. Table 2.7 summarizes the ADLs detected as well as their level of granularity. There were 39 systems in total identified and most used the environment sensors. Only 3 out of the 39 systems used wearables. However, these wearable systems were able to recognize granular activities within "feeding" such as drinking and using cutlery that was not included in the others [60].

Though the intent of the SR was to review technology that is able to detect if an ADL action has been completed and not test if the individual is functionally able to perform the ADL, there may be some correlations with assessment scales that may be drawn. For example, the activities that

technologies in Camp et al.'s review detect are present in the Barthel Index: bathing, dressing, feeding, grooming, mobility, stairs, toileting, and transfer. Sub-activities or finer-grained activities in the Barthel Index, such as activities to define feeding (cutting, or spreading butter), were not included as activities in these systems. According to Camp et al., there are few technologies that are able to recognize fine-skills such as cutlery use which have a "large influence on more general activities such as feeding, and subsequently on overall health" [60]. There also may be concerns with privacy which prevent these technologies from advancing deeper into exploring the detection of these sub-activities. For example, humidity and sound sensors used to detect bathing, a sub-activity of grooming, may pose a privacy risk [60].

The systems identified recognize ADLs by using a series of interactions with the environment sensors and/or wearable sensors. For example, "feeding" can be inferred by activity in the motion sensors in the kitchen/dining room, fridge door sensors, and power sensors attached to a microwave [60]. A full list of these interactions and accompanying ADL can be found in Table 2.8. Though these systems can recognize ADLs and is sufficient for the aim of monitoring functional decline by monitoring a decrease in ADL actions, Camp et al. suggest that methods to "directly [assess] ADL performance" should be considered as it will allow for identification of "exactly where an individual is having difficulty without the reliance on [time-consuming] traditional tests" [60].

Table 2.6: Types of technologies identified in Camp et al's SR [60].

Category	Technology
Environment	Accelerometer
	Barometer
	Door Contact
	Force/Pressure
	Grid-Eye (Infrared array sensor)
	Humidity
	Hydro
	Light
	Motion
	Power Consumption
Wearable	Sound
	Temperature
	RFID Tag
	Wearable Accelerometer
	Wearable Altimeter
Camera	Wearable Barometer
	Wearable Gyroscope
	Wearable Light
	Wearable Temperature
	Depth Camera
	Video Camera

Table 2.7: Types of ADLs detected [60]. For Feeding there was separation in some studies between eating, meal preparation, drinking, and type of meal (breakfast, lunch, and dinner). For Grooming, some studies specified brushing teeth, showering, shaving, and hair styling. Finally for Social Interaction, most studies used the "front door usage," but some specified the time away from home, and the number of visits [60].

ADL Detected
Bed Usage
Dressing
Feeding
Grooming
Household
Medicine
Mobility
Recreation
Sleep
Social Interaction
Stair Usage
Toileting
Transferring
TV Usage

Table 2.8: How to detect each ADL based on the technology found in [60]

ADL Detected	Method
Bed Usage	Pressure sensors or accelerometers attached to the bed
Dressing	Room activity and door sensors attached to specific drawers/wardrobes
Feeding	Room activity, appliance use, or door sensors attached to cupboards within the kitchen area. Wearable sensor data
Grooming	Room activity, changes in temperature/humidity, water usage or specific door usage
Household	Water usage, room activity or appliance usage
Medicine	Door sensors attached to medicine cabinets
Mobility	Room activity or Wearable sensors
Recreation	Room activity or power consumption
Sleep	Room presence (but inactivity), typically the living room & bedroom
Social Interaction	Door sensors attached to the main property entrance
Stair Usage	Combined wearable sensors
Toileting	Room activity, specific door usage, water usage or accelerometers attached to the flush mechanism
Transferring	Room activity or pressure sensors or Wearable sensors
TV Usage	Power consumption/smart switch

2.4.2 Gadey et al's Tech for Monitoring ADL in Older Adults

Compared to Camp et al.'s article which focusses on technology to detect the completion of an ADL, Gadey et al.'s SR focusses more generally on the "extent and diversity" of technologies that can be used for monitoring ADLs in older adults. From 16 articles, the SR yielded 48 technologies with categories of "health-related technology, ambient appliance mounted, ambient location, motion-based, visual or interactive". There is a further category of "ambient" that is not mentioned explicitly in the article but appears as a category. After summarizing Table 4 in Gadey et al.'s article, 26 distinct technologies were found and are listed in Table 2.9 [61]. There may have been additional technologies identified in the SR, but were grouped under more general terms such as "appliance sensors" [61].

There were both Basic ADLs and Instrumental ADLs assessed throughout the 16 articles. In terms of BADLs, walking, transferring, stair climbing, eating, bathing, sleeping, toileting, dressing, and grooming were monitored and assessed. Out of these, "walking" was monitored and assessed the most with 11/16 articles [61]. There were 10 IADLs monitored and assessed: cooking, dishwashing, housework, managing medication, watching TV, phone use, and typing/writing. Out of these IADLs, cooking appeared the most at 6/16 articles [61].

The authors report that within the 16 articles selected wearables and Inertial Measurement Units (IMUs) were the most common technology applied and were used primarily for gait analysis. The accelerometer in the IMU was the most common in the activity recognition, and the entire IMU can be used for detecting postures and activity or inactivity. The article does not go into detail about the algorithms or methods used to detect the ADLs mentioned, but provides the suite of technologies (Wearable+Health, Wearable+Ambient, Wearables alone, etc.) used to detect certain ADLs. However, in the current state, it seems like the technologies and methods available can only detect activity and distinguish between general ADL classes [61]. An ideal monitoring solution would be able to detect changes in living patterns that may indicate early physical or cognitive decline [61]. It is not only the quantity of ADLs that needs to be monitored but also its quality (or how an ADL is being performed).

Table 2.9: Types of technologies identified in Gadey et al's SR [61]. Each system, and not each technology, identified was given categories of health-related technology, ambient appliance mounted, ambient location, motion-based, visual or interactive. This Table is a best effort interpretation of the overlapping categorization of technologies. Where there was overlap (systems containing technology belonging to 2 or more categories), technologies that appeared in at least 2 different systems having the same categorization were grouped into that identifying category.

Category	Technology
Health-Related	Weight Scale BP measurement O2Sat Glucometer Heart Rate ECG Electrooculography Body Temperature Thermometer
Ambient Appliance Mounted	Electricity sensor Flow sensors Door Contact
Ambient Location	Force sensor Motion sensor Magnetic sensor
Ambient	Hygrometer (humidity) Microphones Pressure sensors Force Sensing Insole Barometer Ambient Temperature
Motion-Based	Accelerometer Gyroscope Magnetometer
Visual	Stereoscopic Camera
Interactive	Task List Alerts

2.5 Motivation

Both Camp et al. and Gadey et al. place an emphasis on the monitoring of the quality or performance of ADL. In Camp et al.'s case, doing so would allow a team of healthcare providers to pinpoint "exactly where an individual is having difficulty without the reliance on [time-consuming] traditional tests" [60]. In Gadey et al.'s case, the capability of ADL quality monitoring would allow for the early detection of physical or cognitive decline.

Though there are 6 IADLs that can be focussed on, refer to Table 2.1, the method and technology required to monitor and evaluate each is expected to differ. Also considering the scope and time constraints of this thesis, it is reasonable to approach the problem one ADL at a time. Accordingly, the IADL of cooking or meal preparation was chosen as the focus of this thesis since it presents an opportunity to monitor the quality or performance of an older adult wanting to age-in-place. Cooking is a "cognitive complex task, which requires multiple steps" [62] and coordination between physical and cognitive functionality. Moreover, cooking was identified to be critical in terms of sustaining a healthy lifestyle in older adults [63] and was found to be the highest ranked necessity "for independent living" in interviews with older adults [64]. Though cooking is essential, it is one of the most impacted IADLs after traumatic brain injury (TBI) meaning that decline in ability to cook may be indicative of cognitive decline. Older adults also cook frequently, with 53% of older adults ages 65-80 reported to cook at home 6-7 days a week [65]. Despite the health benefits and enjoyment gained from cooking, cooking is fraught with risks [66]. There are safety concerns with the preparation of a hot meal and it poses risks such as "fires, burns" [64], injuries from improper knife use, and injuries from falls due to an unsafe kitchen environment [66] which monitoring and proactive intervention may have a role in preventing.

In pursuit of monitoring the cooking IADL, the next sections will be dedicated to scoping methodologies for monitoring and evaluating cooking. It will look at how cooking is monitored, the technologies involved, and the accompanying algorithms or models used to evaluate cooking.

2.6 Monitoring Cooking

Before cooking can be assessed, the first step in monitoring cooking is detecting the activity at a macro scale and separating it from all of the other

IADLs and BADLs.

The literature already presents several different ways to detect meal preparation or cooking. Cook achieved this through the use of environment sensors such as motion sensors, door contact sensors, temperature sensors, light sensors, water flow sensors, stove-top burner use sensors and sensors on specific items [67]. The data collected from this system can then be used in a Naive Bayes Classifier, Hidden Markov Model (HMM), or Conditional Random Field, which are traditionally "robust in the presence of moderate amount of noise, are designed to handle sequential data, and generate probability distributions over the class labels" [67], to classify the activities. Cook also explored using the output probability distributions of each of these models as inputs to a boosted tree classifier to improve classification accuracy [67]. Logan and Healy used a modified form of AdaBoost with simple linear weak learners to distinguish meal preparation and eating through accelerometer, video capture, and audio capture data [68]. Sarma et al. used Long short-term memory (LSTM) to determine various ADLs from datasets containing motion sensors, door closure sensors and temperature sensors data [69]. Chibaudel et al. detected cooking by noting the physical location of the participant and their refrigerator usage through door sensors placed in the refrigerator and motion sensors in the kitchen [70]. Yordanova et al. used Decision Trees (DT), Computational Causal Behavior Models (CCBM), and HMM to process data from the full-scale SPHERE Smart Home system consisting of temperature sensors, humidity sensors, luminosity and motion sensors, water and electricity usage sensors, cameras, and door contact sensors to classify the preparation of a wide range of recipes as "cooking" [71].

Although there have been many studies involving the detection of the cooking ADL, few studies assess the quality of cooking and provide feedback to the user. One attempt at assessing quality involved quantifying the number of departures from a given task. Cook and Schmitter-Edgecombe used motion sensors, analog sensors for water and stove usage, open/shut sensors for the status of cabinets, and load sensors for the absent/present status of items to assess the quality of ADLs including meal preparation. If a correct sequence of tasks was done and if the task was done efficiently, then an activity is considered normal. Any significant departures from this "correct sequence" and normal time required to complete each step was only left with a tag of "anomalous" [72]. There is no further information with respect to how much of an anomaly the error was or what to do about it. Similarly, in Menghi et al.'s study, errors were identified in a series of tasks, but nothing

was done to evaluate the severity of the error and no feedback was given to the user [70].

To address these shortcomings, Dawadi et al. performed an exploratory study to map smart-home sensors readings to ADL performance ratings based on a custom scale (Figure 2.8). The criteria for performance was based on whether "important steps are skipped, performed out of sequence or performed incorrectly" (such as forgetting to close the fridge or leaving on a burner) [73]. Tasks completed inaccurately or inefficiently may be indicative of a cognitive health condition [73]. 8 activities were selected including sweeping and preparing microwavable soup. After completing each activity, each participant was rated by two observers based on the scale in Figure 2.8. Features were extracted from the sensor readings obtained during each activity such as duration of activity, sensor count, unrelated sensors, door sensor count, item sensor count, etc. Then, Support Vector Machines (SVMs) were trained on the features extracted and observer score labels to predict a score from sensor reading inputs. The authors note that though predicting task quality based on smart home sensor readings is possible, there is only moderate correlation between the predictions and the observer ratings [73]. The authors take the evaluation one step further and attempt to classify the cognitive health of the participants as cognitively healthy (CH), mild cognitive impairment (MCI) and dementia (D) based on the performance scores. Using SVM trained on the performance score and health status of the participant, the AUC score was calculated. For MCI vs. D was 0.56 and for D vs. CH it was 0.72 when considering all activities. This means that the classifier was almost no better than random guessing for the MCI vs. D case but had better predictability for the D vs CH case. Though it is possible to perform a "limited health assessment" with task performance, the authors mention that the accuracy may be improved with additional tasks or features obtainable from sensors that can provide "finer resolution" such as wearable sensors [73].

Based on the articles reviewed thus far, detecting the cooking activity alone may not be indicative of ADL cooking quality. However, detecting subtasks within the cooking activity and evaluating metrics or features such as steps skipped, performed out of sequence, and duration of these subtasks can be correlated with a performance score (like in Dawadi et al. study [73]). Although Dawadi et al.'s approach was able to predict performance scores and distinguish between cognitively healthy and dementia patients, there are potentially two limitations. First Dawadi's sensor suite provides only a

Score	Criteria
1	Task completed without any errors
2	Task completed with no more than two of the following errors: non-critical omissions, non-critical substitutions, irrelevant actions, inefficient actions
3	Task completed with more than two of the following errors: non-critical omissions, non-critical substitutions, irrelevant actions, inefficient actions
4	Task incomplete, more than 50% of the task completed, contains critical omission or substitution error
5	Task incomplete, less than 50% of the task completed, contains critical omission or substitution error

Figure 2.8: Scoring Criteria for Dawadi et al.’s experiment.

coarse resolution of the appliance/item sensors which limited the subtasks that could be detected to item and appliance interactions. Thus, Dawadi et al. proposed searching for methods/sensors that may provide finer resolution such as wearables [73] as a next step. Second, features extracted from the sensor readings were used to predict performance scores that were custom-made. Though the scale is reasonable for evaluating performance, it may not be as well-validated as a scale devised in the observational traditional assessments for ADLs.

Following Dawadi et al.’s suggestion to search for methods/sensors that can detect finer-grained activities, there have been other attempts at the classification of the subtasks of cooking and other ADLs:

- Chen et al. used 5 IMUs each containing an 3D accelerometer and 3D gyroscope. 2 were placed at each wrist, 2 on each upper arm and one on the hip. These IMUs were used to detect 19 ADL subtasks: brush teeth, mixing powders, spreading butter/jam, eating with hands, eating with utensils, buttoning shirt/coat, put and take off the coat, moving items horizontally, reaching up, reaching down, washing dishes, chopping, pan stirring, serve on a plate, sweeping, vacuuming, wiping horizontal surface, wiping vertical surface, and folding clothes [74]. From each axis of each sensor, 3 second windows were created and 4 statistical features were extracted: standard deviation, mean, slope (from linear regression), and auto-correlation (for repetitive actions). For feature selection, the authors used XGBoost since XGBoost has ”better effi-

ciency and controls for overfitting” [74] compared to the other models. After the features were selected, the Decision Tree, Random Forest, SVM, and XGBoost were tested to obtain the optimal model based on accuracy, recall, precision, and ROC AUC [74].

- Pan et al. detected fine-grained actions using a combination of structural vibration sensing and single-point electrical load sensing [75]. 8 activities were classified: kettle use, stove use, put on stove, use microwave, open/close microwave door, put down food, use vacuum, and footsteps. Segments of the signal from both types of sensors with significantly higher energy were extracted as events. Then, from each sensor within each event segment, the power spectral density was calculated, concatenated together and used as a feature vector [75]. SVM was chosen as the model for classification of the 8 activities mentioned previously.

2.7 Next Steps

Though there was some degree of success in the methods that the previous articles used to detect subtasks within ADLs, there may be some limitations. In Chen et al.’s approach, there may be too many sensors involved and may pose a challenge for older adults to put on and take off every day for continuous monitoring. Pan et al.’s usage of vibrational sensors is novel, but by design of environment sensors which continuously run, there may be ethical issues with control (autonomy) and the older adult will not be able to freely turn the system off. Furthermore, Pan et al.’s models are trained only on able-bodied individuals meaning there is no indication of performance on individuals with impairments. Judging from the limitations present in the articles scoped out so far, a first step in the endeavor of monitoring and evaluating cooking in older adults may be to develop a method that is practical, accurate and robust to variation caused by impairments in detecting these subtasks in real-time.

Toward the development of a practical and accurate method for detecting cooking subtasks, it is hypothesized that a positional + IMU sensor placed on the dominant wrist in a wrist-watch form-factor would satisfy these ethical, practical and functional requirements. The older adult will be in full control of the system and can choose to not be monitored by taking off this watch

satisfying autonomy [76]. Also, no cameras or video capture is used, alleviating concerns with privacy [77]. In regard to practicality, there is only one wearable involved and not 5 [74] making it easy for the older adult to wear the system. Finally, since certain subtasks occur only within certain areas in the kitchen, it is expected that the addition of the positional data will reduce misclassification from confounding motions in different locations that IMUs alone may not be able to separate. Positional data will also provide contextual information about interaction with appliances, or items within the kitchen without the need to instrument every item single item and appliance in the kitchen. For example, the opening action occurring near the fridge is indicative of opening the fridge as opposed to at the cabinet which means that the older adult opened the cabinet.

The rest of this thesis will work toward validating the performance and accuracy of the proposed IMU + positional sensor system for detecting cooking subtasks. Like in Chen et al's study, datasets of "atomic actions" or singular subtasks such as chopping, pan stirring, and washing dishes [74] will be created from an able-bodied individual with simulated impairments (eg. blind-fold to simulate visual impairments [78]). Feature extraction and feature selection will be performed to obtain a best set of features for predicting a subtask. Multiple models will be trained and evaluated for time-series classification including but not limited to, Random Forests, SVM, Convolutional Neural Networks (CNN) for stacked time series and LSTM. Subsequently, goal-based tasks, such as making a sandwich or preparing canned soup, which contain a collection of the aforementioned subtasks will be performed (normally and with simulated impairments) and used to validate the trained model. However, prior to this classification task there are 2 items that must be addressed: first, testing the proposed system and characterizing performance such as accuracy, and second determining the subtasks and goal-based tasks to be used for this thesis.

Chapter 3

System Testing and Tuning at the Independent Living Suite

3.1 System Tuning Review

The Pozyx Creator Kit comes with anchors and several tags. Anchors are mounted on the walls and are used to position the tags. Multiple tags may be positioned at the same time. The Pozyx Creator kit uses ultrawideband (UWB) signals with the two-way ranging protocol to localize the tag. The tag is mounted on custom 3D printed wearables which the participant can wear as a wrist-watch or a necklace. Through trial-and-error and consultation with the Pozyx Creator Documentation [79, 80] it was determined that the accuracy of the system depends on factors listed below:

- Number of anchors
- Position of anchors

These variables were modified to achieve satisfactory actual position error and standard deviation below the expected error of 30 cm for UWB systems. The protocol for obtaining data and evaluating the actual position error and standard deviation is described in the next section.

3.2 Methodology

This protocol tests the X, Y, and Z positional accuracy of the Pozyx Creator system in the Independent Living Suite (ILS) at the Glenrose Rehabilitation Hospital by having a participant stand at a specific location in each room. Permanent appliances or furniture such as the stove or dining table were used as much as possible to ensure that the experiment is repeatable.

3.2.1 Setup

Masking tape was used to mark the locations where the participant should place their feet. The following procedure was followed to place the tape:

1. Using a measuring tape, measure 1 meter out from the middle of the appliance or furniture and place a 20 cm piece of tape centered on, perpendicular to and underneath the measuring tape (the tips of the participant's toes should be 1 meter away from the appliance).

2. Place parallel tape on the sides of the tape placed in Step 1 to constrain the feet to a box. (The participant should have their toes on the tape perpendicular to the measuring tape and usually facing the appliance or furniture). Figure 3.1 outlines some examples of tape placements.



Figure 3.1: Box tape placement at the stove, fridge, and dining table. Participant's toes and sides of feet should touch the tape.

Following the tape placement guidelines outlined at the beginning of this section, tape was placed at or near the following locations. Refer to the AUTOCAD floor plan for the location of the rooms (Figure 3.2):

- The Hallway between Living Room and Kitchen facing the Dining Table.
- The Living Room facing the Desk.
- The Bedroom facing the bed.
- The Hallway between the bedroom and the bathroom, facing away from the wall.

- Bathroom facing the toilet.
- Kitchen facing the stove.

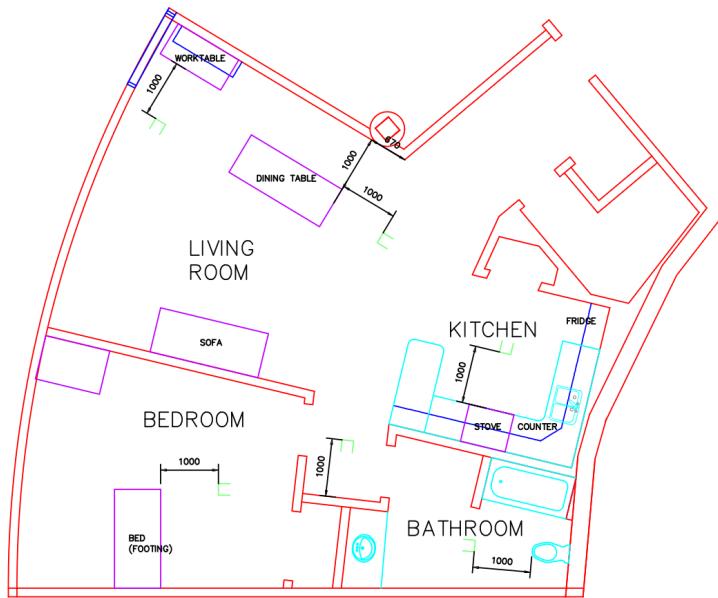


Figure 3.2: Floor plan of the ILS. Positions where the participant stood 1 meter from appliances or furniture are marked in the green "open" box on the floorplan.

3.2.2 Protocol

A stopwatch python script was created with predetermined labels and used as the ground truth for positions. A single participant wore the tag on a 3D printed necklace mount (Figure 3.3). The measuring tape was used to measure the height from the ground and height when squatting. For this participant, the standing height was **144cm** and the squatting height was **68.5cm**.



Figure 3.3: The Pozyx tag mounted in a custom 3D printed necklace mount.

The protocol had the following steps:

1. At the first location (Hallway Between Living Room and Kitchen) stand still for 10 seconds
2. Squat still for 10 seconds.

3. Move to next position.
4. Repeat steps 1-3 until all of the positions have been reached.
5. Finally return to the first position (Hallway Between Living Room and Kitchen)

There were 3 trials for each configuration. Following the guidelines from the Pozyx Creator Setup [80] anchors were staggered at heights of 1.4m and 2.4m (ceiling height) for 4, 5, 6, 8, and 10 anchors. A configuration where anchors were all low (10cm) were tested for 8 anchors and configurations where anchors were all high (2.4m) were tested for 8 and 9 anchors.

3.3 Results

Trials for each configuration were aggregated, transition periods were removed, data of interest was time normalized and the error and standard deviation of each location while standing and squatting were calculated. An as-built AUTOCAD file of the ILS was used to obtain the real position and used in the calculation of the error between measured versus the actual position. The results of the experiment are summarized in the heatmap tables (Figures 3.4, 3.5 and 3.6) with a minimum darkness set at 30cm and a maximum darkness set at 60cm.

	X Position Error at Each Location (cm)							
	POS_X_A4	POS_X_A5	POS_X_A6	POS_X_A8	POS_X_A10	POS_X_A8H	POS_X_A8L	POS_X_A9H
Go Hallway between kitchen and living	40.1	41.5	42.2	47.9	33.0	11.8	3.5	28.0
Go Hallway between kitchen and living(sit)	18.6	11.7	7.4	44.2	20.6	29.2	21.9	41.4
Living Room	50.4	57.8	48.3	7.7	34.5	20.0	31.3	41.8
Living Room(sit)	44.6	30.2	20.0	28.8	23.7	5.2	6.5	11.0
bathroom	49.1	39.8	33.0	65.3	15.4	111.9	120.6	52.0
bathroom(sit)	29.2	19.6	22.9	23.2	39.0	64.5	24.6	61.1
bedroom	24.1	2.5	39.4	58.3	55.8	99.8	78.3	49.2
bedroom(sit)	26.9	12.8	4.1	103.8	45.9	53.5	27.0	37.5
hallway between bedroom and bathroom	2.7	14.9	5.3	5.4	7.6	6.6	3.7	36.1
hallway between bedroom and bathroom(sit)	2.8	13.9	26.5	0.1	3.4	1.7	51.3	22.2
kitchen	23.7	26.8	18.5	30.1	31.5	37.3	29.1	7.7
kitchen(sit)	8.2	22.2	8.8	41.9	31.2	43.7	44.1	28.0

(a)

	X Standard Deviation at Each Location (cm)							
	POS_X_A4	POS_X_A5	POS_X_A6	POS_X_A8	POS_X_A10	POS_X_A8H	POS_X_A8L	POS_X_A9H
Go Hallway between kitchen and living	105.9	56.5	62.1	39.2	42.2	43.8	29.1	28.2
Go Hallway between kitchen and living(sit)	9.8	19.2	12.7	24.5	13.7	12.3	20.0	15.6
Living Room	5.4	7.9	6.4	18.7	19.1	17.3	12.0	12.9
Living Room(sit)	11.6	24.9	19.2	18.0	20.2	17.4	16.2	13.2
bathroom	44.4	27.3	19.6	82.5	43.4	65.3	90.6	39.1
bathroom(sit)	13.8	12.6	9.7	68.0	39.7	44.5	80.2	23.1
bedroom	17.7	17.3	17.6	21.0	21.6	36.3	30.5	24.7
bedroom(sit)	7.9	20.7	28.1	28.3	14.6	29.5	29.1	14.4
hallway between bedroom and bathroom	10.2	12.5	6.9	24.6	18.1	16.0	13.6	19.3
hallway between bedroom and bathroom(sit)	14.6	4.8	10.8	13.0	13.4	10.1	12.3	12.9
kitchen	16.7	7.8	10.3	5.1	5.0	7.9	7.9	7.8
kitchen(sit)	20.7	6.6	16.4	14.4	9.6	8.5	9.7	5.1

(b)

Figure 3.4: The positional error in X (a) and the standard deviation in X (b) at each location and body position

	Y Position Error at Each Location (cm)							
	POS_Y_A4	POS_Y_A5	POS_Y_A6	POS_Y_A8	POS_Y_A10	POS_Y_A8H	POS_Y_A8L	POS_Y_A9H
Go Hallway between kitchen and living	20.5	28.7	38.8	95.4	62.3	58.2	8.6	28.5
Go Hallway between kitchen and living(sit)	46.5	58.2	57.9	65.5	49.5	63.4	31.7	42.7
Living Room	11.6	6.2	3.2	62.3	63.5	47.1	37.4	38.0
Living Room(sit)	7.8	0.7	3.7	47.8	35.6	25.8	32.7	15.1
bathroom	27.9	0.4	7.8	44.4	56.1	47.3	10.3	38.2
bathroom(sit)	54.7	39.2	37.7	53.7	29.7	5.1	15.1	1.0
bedroom	22.8	14.5	9.2	39.5	16.1	71.7	58.4	5.7
bedroom(sit)	20.2	5.3	33.9	60.9	5.0	34.6	1.8	8.7
hallway between bedroom and bathroom	4.6	7.8	24.8	7.1	9.4	23.5	2.0	19.4
hallway between bedroom and bathroom(sit)	56.2	67.0	44.4	31.0	7.7	28.3	8.8	13.8
kitchen	16.8	2.8	10.8	28.5	33.7	39.0	38.0	31.3
kitchen(sit)	17.5	4.0	9.0	21.9	31.1	15.5	28.6	22.2

(a)

	Y Standard Deviation at Each Location (cm)							
	POS_Y_A4	POS_Y_A5	POS_Y_A6	POS_Y_A8	POS_Y_A10	POS_Y_A8H	POS_Y_A8L	POS_Y_A9H
Go Hallway between kitchen and living	103.7	53.5	62.1	115.5	79.0	137.9	25.8	24.7
Go Hallway between kitchen and living(sit)	10.3	13.7	10.7	23.6	17.8	21.1	27.6	16.0
Living Room	5.7	8.3	4.8	13.7	16.1	17.4	20.8	14.2
Living Room(sit)	11.1	14.7	11.0	21.3	12.2	16.3	13.1	7.5
bathroom	35.1	25.1	21.9	50.8	24.8	36.2	39.8	23.4
bathroom(sit)	29.8	42.1	29.3	19.1	23.4	20.3	38.9	17.8
bedroom	15.6	17.7	14.2	18.9	12.2	18.4	23.7	9.5
bedroom(sit)	9.5	12.4	14.3	24.9	14.9	16.8	17.5	6.2
hallway between bedroom and bathroom	15.2	9.0	12.2	23.0	13.6	13.4	9.5	13.0
hallway between bedroom and bathroom(sit)	12.7	10.5	16.7	8.0	12.9	12.3	12.1	8.2
kitchen	17.6	9.1	6.2	9.9	7.1	14.0	9.0	9.8
kitchen(sit)	24.4	16.0	17.2	15.3	11.0	12.7	16.1	5.9

(b)

Figure 3.5: The positional error in Y (a) and the standard deviation in Y (b) at each location and body position

	Z Position Error at Each Location (cm)							
	POS_Z_A4	POS_Z_A5	POS_Z_A6	POS_Z_A8	POS_Z_A10	POS_Z_A8H	POS_Z_A8L	POS_Z_A9H
Go Hallway between kitchen and living	132.3	56.9	47.9	180.1	87.6	7.5	58.2	3.2
Go Hallway between kitchen and living(sit)	137.0	207.8	184.0	232.3	104.7	7.6	22.8	16.0
Living Room	121.5	36.7	41.7	99.4	88.5	58.1	54.8	57.0
Living Room(sit)	79.1	177.1	197.1	224.9	16.7	30.8	97.0	17.2
bathroom	84.0	66.7	114.7	64.2	27.3	43.6	147.6	80.3
bathroom(sit)	43.2	10.2	11.2	211.1	66.4	26.3	4.6	66.2
bedroom	57.5	65.8	121.6	192.9	233.5	155.3	234.6	37.1
bedroom(sit)	39.7	1.3	54.8	305.5	3.5	80.9	142.5	20.6
hallway between bedroom and bathroom	74.0	103.4	4.1	208.3	48.6	30.8	72.3	71.7
hallway between bedroom and bathroom(sit)	33.4	26.2	59.1	84.2	2.8	61.1	54.4	65.2
kitchen	70.5	117.0	40.5	0.1	38.7	33.5	73.8	31.5
kitchen(sit)	119.0	3.9	25.9	216.1	70.6	38.5	139.6	20.5

(a)

	Z Standard Deviation at Each Location (cm)							
	POS_Z_A4	POS_Z_A5	POS_Z_A6	POS_Z_A8	POS_Z_A10	POS_Z_A8H	POS_Z_A8L	POS_Z_A9H
Go Hallway between kitchen and living	93.9	80.0	89.6	68.1	69.9	58.7	110.3	33.8
Go Hallway between kitchen and living(sit)	84.8	110.0	123.2	151.2	108.1	21.0	77.3	9.0
Living Room	46.7	36.1	23.5	92.1	25.6	20.0	144.2	17.1
Living Room(sit)	53.1	118.1	112.8	107.5	106.7	13.2	87.2	22.0
bathroom	89.9	105.2	69.6	99.9	77.3	79.4	151.9	40.7
bathroom(sit)	107.7	18.3	35.4	90.0	115.6	119.0	93.6	63.7
bedroom	65.4	109.7	67.8	142.2	58.5	24.2	246.9	20.0
bedroom(sit)	15.1	28.4	66.1	154.6	130.0	25.6	144.2	12.1
hallway between bedroom and bathroom	37.2	50.0	21.1	51.5	98.9	31.1	150.1	30.2
hallway between bedroom and bathroom(sit)	104.2	13.2	24.7	176.0	112.5	14.1	57.9	15.7
kitchen	75.2	62.0	64.5	61.8	13.2	24.4	26.3	17.0
kitchen(sit)	149.3	46.5	95.5	172.8	119.5	15.9	42.7	7.2

(b)

Figure 3.6: The positional error in Z (a) and the standard deviation in Z (b) at each location and body position

3.4 Discussion

3.4.1 X Position

Visually, the heatmap of error in the X position shows different spots where the system struggled to obtain the location based on the AUTOCAD as-builts depending on the configuration selected. For 4, 5 and 6 anchors, the errors seemed to be larger in the living room and the hallway between the kitchen and the living room. 8, 10, 8 (L)ow, 8 (H)igh, and 9H anchors seemed to struggle most around the bathroom and bedroom area. The standard deviation in X position seems to follow a similar pattern where 4, 5 and 6 anchors have higher standard deviation in hallway between the kitchen and living area and 8, 10, 8 (L)ow, 8 (H)igh, and 9H seems to struggle the most in the bathroom. Out of all of the configurations the 9H configuration has the most locations where the standard deviation is acceptable.

3.4.2 Y Position

For 4, 5, and 6 anchors, the error seems to increase in the seated position meaning that there may be some dependence on the Z position. This occurs in the hallway between the kitchen and the living room, the bathroom and the hallway between the bedroom and bathroom. There seems to be a large struggle for 8, 10 and 8H anchors to pinpoint the Y position in the hallway between the kitchen and the living, the living room and the bathroom. The 8L anchor configuration struggled when in the bedroom, but was overall within or near the acceptable threshold of 30cm. 9H anchors overall seemed to be the best at determining the Y position with mild errors at the hallway between the kitchen and the living room, the living room and the bathroom.

In terms of standard deviation, anchor configurations 4, 5, 6, 8, 8H, and 10 had trouble at a height of 144cm, but otherwise had low standard deviation. 8L had minor issues regarding standard deviation in the bathroom but was otherwise low. The 9H configuration seemed to yield the lowest standard deviations in the Y Position.

3.4.3 Z Position

The Z position at many of the locations and all configurations seem to deviate from the measured heights and have high error. Only the 8H and 9H

configurations have acceptable standard deviations for most of the rooms (there is still some struggle in the bathroom). Considering the inaccuracies in the Z positioning, it is recommended that the Z not be used as a absolute source of truth for height. Rather Z position should be used relative to another reference tag with the 9H configuration. For example, a necklace tag may be combined a wrist tag. When standing, the position of the wrist may be compared with the position of the necklace to determine if the wrist is above, below or at chest height.

3.4.4 Overall

The 9H configuration seems to provide the most reliable data when observing the standard deviations of the X, Y, and Z positions. With this configuration, each room had around 4 anchors surrounding it Figure 3.7

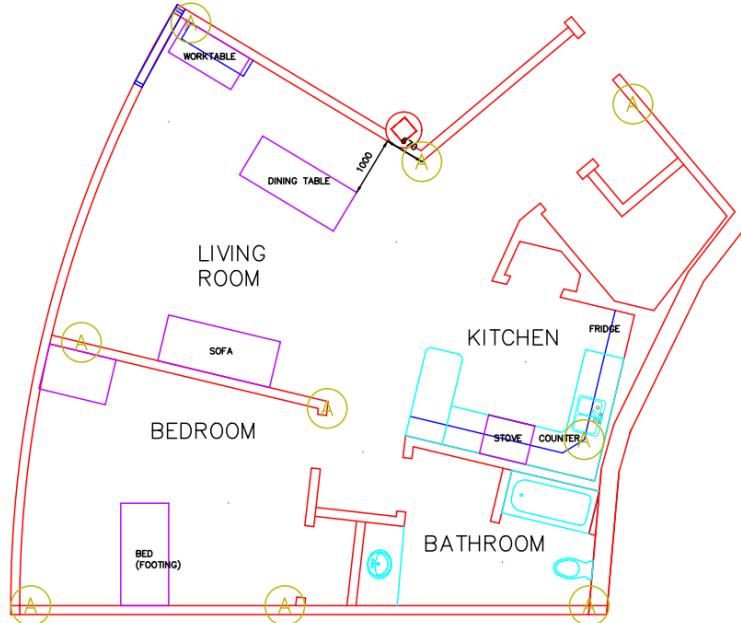


Figure 3.7: ILS Floorplan with the 9 anchors all high.

Though the inaccuracies in the hallway, living room, bathroom and bedroom may prevent the 9H configuration from using heuristics for classification

at these locations, the inherent repeatability evident in the low standard deviation in each axes of the position can make the position data from the 9H configuration a candidate for machine-learning based classification.

Chapter 4

Classification of Time Series Data

Based on the findings in Chapter 3, it was determined that Machine Learning (ML) would be a suitable method for classifying the data obtained from the Pozyx system. Prior to attempting any experimentation, this chapter will conduct a survey of the current ML techniques available for the classification of Time Series Data. These techniques can be split into the Classic ML methods and Neural Networks.

4.1 Classic Machine Learning Methods

4.1.1 k-Nearest Neighbours (kNN)

Overview

The k-Nearest Neighbours is a classifier that assigns a class to an unlabeled observation by looking at the class of k neighbours and choosing the class that appears the most within these k neighbours. The "nearest neighbours" are determined through a measure of Euclidean-Distance D between a neighbour p and unlabeled point q with n features is shown in Equation 4.1 [81]

$$D(p, q) = \sqrt{(p_1 - q_1)^2 + (p_2 - q_2)^2 + \dots + (p_n - q_n)^2} \quad (4.1)$$

Using Equation 4.1 the top k neighbours with the lowest distance are considered and the class that occurs the most within these nearest neighbours is the class chosen for the unlabeled observation. In the event that there is a tie between the classes, resolution of the tie depends on the implementation of the library used. In R `kNN()` resolves ties by picking a random tied candidate class [81] and scikit-learn's `kNeighborsClassifier` uses `scipy`'s "mode" method which returns the first class that ties in the array [82].

Dynamic Time Warping (DTW) as a Distance

Though Euclidean-Distance may be suitable for problems with a fixed feature set, or vectors with the same length, patterns that manifest in time-series data may be stretched, compressed, or shifted in the temporal domain. Using Euclidean distance, a signal compared with the same signal shifted $t + 1$ may result in large distances even though the signals compared were the same signals [83]. A method called Dynamic Time Warping (DTW), was devised to counter the shortcomings of Euclidean distance and is robust to temporal variation that occurs naturally. It achieves this by calculating all

the distances from one point to every other point and chooses a path that costs the least, finding the minimum distance between two vectors that do not have to be the same length. Mathematically, for $X = (x_1, x_2, \dots, x_n)$ and $Y = (y_1, y_2, \dots, y_m)$ be two time series of length n and m respectively, DTW first begins with calculating the Cost Matrix which is the cost between each pair in the time series [83]

$$\forall_i \in 1, \dots, n, \forall_j \in 1, \dots, m, C_{ij} = f(i, j) \quad (4.2)$$

Where f is the squared Euclidean-Distance (equation 4.1 squared) for multivariate time-series [83].

$$f(x, y) = D(x, y)^2 \quad (4.3)$$

A warping path is defined as $p = (p_1, \dots, p_L)$ through the cost matrix C begins on $p_1 = (1, 1)$, must end on $p_L = (n, m)$ and moves with step $p_{l+1} - p_l \in \{(1, 0), (1, 1), (0, 1)\}$. The costs in the along the paths are summed And the DTW score is the path that costs the least [83].

$$DTW(X, Y) = \min_{p \in P} C_p(X, Y) \quad (4.4)$$

Though every path does not need to be calculated as dynamic programming can be leveraged, the time complexity is still high at $O(nm)$ [83] and this time complexity must be considered when designing a real-time classification system.

Tunable Parameters

The tunable parameter in kNN is k and controls the number of neighbours selected. The optimal k should be empirically chosen by varying k for the selected dataset. When using kNN, the entire training dataset is used as-is. Therefore, class imbalance should also be considered. For example, a higher number of class A than class B could lead to a higher density of class A. Even though the unlabeled observation may be closer to class B, the classifier may choose class A because class B has "run out" of neighbours. With increasing k , the inaccuracy caused by class imbalance becomes even more apparent. If k exceeds the number of samples in the class with the lower number of samples, then the class with the higher number of samples will always be chosen.

4.1.2 Support Vector Machines (SVM)

Overview

Support Vector Machines (SVM) involves finding a hyperplane that best separates 2 classes. There are two cases, the linearly separable case and the non-linearly separable case. In the linearly separable case, the hyperplane perfectly separates the two classes (see Figure 4.1). The idea is to maximize the "margin" which is the training data that is closest to the hyperplane [84]. Cervantes et al. depict this as maximizing the distance between parallel hyperplanes H1 and H2 (which pass through the "support vector" data-points of the training dataset) in Figure 4.1 to optimize the generalization capability of the model.

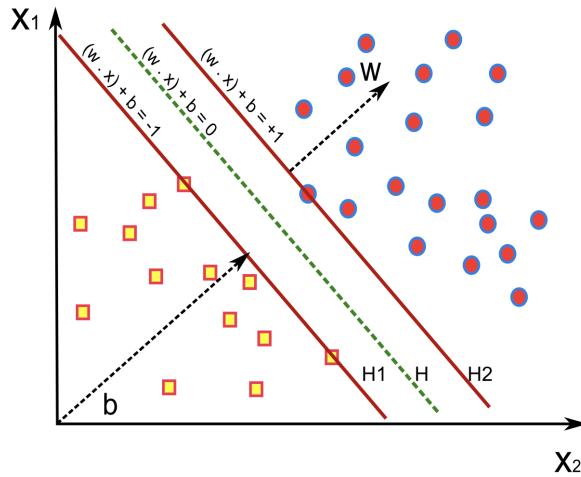


Figure 4.1: SVM Best Hyperplane in the linearly separable case [84].

The linearly separable case is rare in real life. Instead, datapoints from one class may seem to mix with the other class at the optimal boundary (Figure 4.2). To still find this optimal boundary, a positive slack factor ζ_i is introduced and controlled by a parameter C that controls the width of the margin. A lower C allows for a wider margin that may improve generalizability in real life at the cost of more misclassification of the training dataset, whereas a higher C tightens the margin and minimizes the classification errors in the training dataset but may decrease generalizability in real life [84].

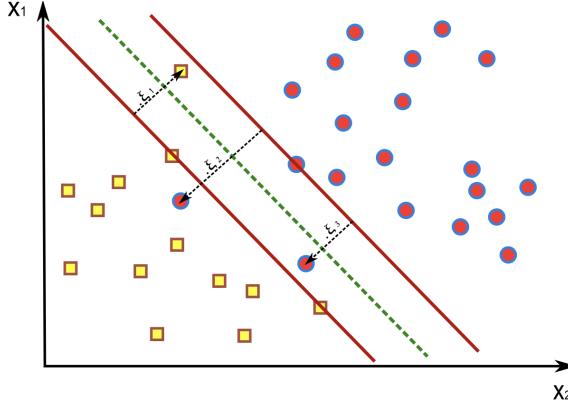


Figure 4.2: SVM Best Hyperplane in the non linearly separable case [84].

SVM can also be used to separate data that is not separable by a linear hyperplane (Figure 4.3) by using a kernel. Kernels reframe the problem in a highly dimensional space called the "feature space" and in this feature space, it is simple for the algorithm to find the hyperplane. In the next subsection that will detail the convex optimization problem for SVM, it is found that the function that needs to be optimized depends on the inner product between every sample $\langle x_i \cdot x_j \rangle$. Transforming the original dataset to this feature space through a transfer function ϕ yields an optimization problem that depends on $\langle \phi(x_i) \cdot \phi(x_j) \rangle$. Kernel functions $K(x_i, x_j)$ are special functions that provide an equivalent way to calculate $\phi(x_i) \cdot \phi(x_j)$ (ie $\phi(x_i) \cdot \phi(x_j) = K(x_i, x_j)$) without having to initially transform the original dataset using ϕ since transforming the dataset to the feature space may be a costly operation if there are many features. The following are popular kernels [84]:

1. Linear Kernel: $K(x_i, x_j) = x_i \cdot x_j$
2. Polynomial Kernel: $K(x_i, x_j) = (1 + x_i \cdot x_j)^p$
3. Gaussian Kernel: $K(x_i, x_j) = e^{-\frac{\|x_i - x_j\|^2}{2\sigma^2}}$
4. RBF Kernel: $K(x_i, x_j) = e^{-\gamma(x_i - x_j)^2}$
5. Sigmoid Kernel: $K(x_i, x_j) = \tanh(\eta x_i \cdot x_j + v)$

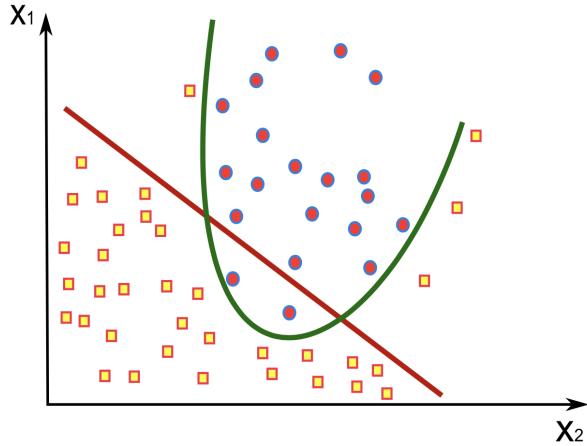


Figure 4.3: SVM non-linear classifier (green line) [84].

Since SVM can only separate between 2 classes as a binary classifier, a further step must be done to separate multiple classes. Scikit-learn uses a one vs. rest strategy by default. For each class a decision boundary between itself and the rest of the samples are created. For an unlabeled point a probability of membership to each class is calculated and an argmax function is used to determine the membership [85].

Theory of SVM: Linearly Separable

Consider a training dataset $X = \{x_i, y_i\}_{i=1}^n$ consisting of n samples of feature vectors $x_i \in \mathbb{R}^d$ and labels $y_i \in \{-1, 1\}$. In the linearly separable case of SVM, the two classes in our training dataset can be separated perfectly by a hyperplane. The data labelled with class $y = 1$ will be on one side of the hyperplane and the other class $y = -1$ will be on the other side of the hyperplane. The equation for a hyperplane is given in Equation 4.5

$$\mathbf{w} \cdot \mathbf{x} + b = 0 \quad (4.5)$$

Where $x \in \mathbb{R}^d$, and \mathbf{w} the weights for the hyperplane and b the bias. For simplicity of visualization, consider $x \in \mathbb{R}^2$. The SVM optimization problem tries to find \mathbf{w} and b such that the margin (ie. the distance to the closest point(s) to the hyperplane) is maximized. One variation of this margin called the functional margin F uses the y_i label to ensure that the margin is positive for individual samples correctly classified by some hyperplane with parameters (\mathbf{w}, b) and is given by the Equation 4.6

$$F = \min_{i=1 \dots n} [y_i(\mathbf{w} \cdot x_i + b)] \quad (4.6)$$

In the SVM optimization problem, this functional margin is set to $F = 1$ as shown in 4.1 and results in the parallel hyperplanes H1 and H2 as the constraint of the problem (Eq. 4.7).

$$y_i(\mathbf{w} \cdot x_i + b) \geq 1 \forall i \quad (4.7)$$

The goal now is to maximize the distance between the hyperplanes H1 and H2 to obtain the optimal hyperplane H which falls in the middle of H1 and H2. The minimum distance from H to H2 (or H to H1) is called the geometric margin. This minimum distance can be found by taking the vector between any point on H2: p_{H2} and any point on H1: p_{H1} and projecting it onto the unit normal vector of H1 or H2 given by \mathbf{w} . Keep in mind that p_{H2} must satisfy $\mathbf{w} \cdot \mathbf{x} + b = 1$ and p_{H1} must satisfy $\mathbf{w} \cdot \mathbf{x} + b = -1$

$$\begin{aligned}
d &= \|\text{proj}_{\mathbf{w}}(p_{H2} - p_{H1})\| \\
&= \left\| \left((p_{H2} - p_{H1}) \cdot \frac{\mathbf{w}}{\|\mathbf{w}\|} \right) \frac{\mathbf{w}}{\|\mathbf{w}\|} \right\| \\
&= \left\| (p_{H2} \cdot \mathbf{w} - p_{H1} \cdot \mathbf{w}) \frac{\mathbf{w}}{\|\mathbf{w}\|^2} \right\|
\end{aligned}$$

p_{H2} must satisfy $\mathbf{w} \cdot p_{H2} + b = 1$ (ie $\mathbf{w} \cdot p_{H2} = b - 1$) to be on hyperplane H2 and p_{H1} must satisfy $\mathbf{w} \cdot p_{H1} + b = -1$ to be on hyperplane H1 (ie $\mathbf{w} \cdot p_{H1} = -b - 1$)

$$\begin{aligned}
&= \left\| ((1 - b) - (-b - 1)) \frac{\mathbf{w}}{\|\mathbf{w}\|^2} \right\| \\
&= \left\| 2 \frac{\mathbf{w}}{\|\mathbf{w}\|^2} \right\| \\
d &= \frac{2}{\|\mathbf{w}\|}
\end{aligned} \tag{4.8}$$

To find the optimal hyperplane, the distance $d = 2/\|\mathbf{w}\|$ must be maximized or equivalently $\|\mathbf{w}\|^2$ (which is a convex function [84]) can be minimized giving rise to the following convex optimization problem that can be solved using Lagrange Duality. The primal problem is:

$$\begin{aligned}
&\min \|\mathbf{w}\|^2 \\
\text{s.t. } &y_i(\mathbf{w} \cdot x_i + b) \geq 1 \quad \forall i
\end{aligned} \tag{4.9}$$

Converting equation 4.9 to the Lagrange Formulation which causes the constraint to move to the objective function and act as a penalty if the constraint is violated:

$$L(\mathbf{w}, b, \alpha) = \frac{1}{2} \langle \mathbf{w} \cdot \mathbf{w} \rangle - \sum_{i=1}^n \alpha_i [y_i(\langle \mathbf{w} \cdot x_i \rangle + b) - 1] \tag{4.10}$$

Setting partial derivatives with respect to \mathbf{w} and b equal to zero to minimize the Lagrangian:

$$\begin{aligned}\frac{\partial L(\mathbf{w}, b, \alpha)}{\partial \mathbf{w}} &= \mathbf{w} - \sum_{i=1}^n \alpha_i y_i x_i = 0 \rightarrow \mathbf{w} = \sum_{i=1}^n \alpha_i y_i x_i \\ \frac{\partial L(\mathbf{w}, b, \alpha)}{\partial b} &= -\sum_{i=1}^n \alpha_i y_i = 0 \rightarrow \sum_{i=1}^n \alpha_i y_i = 0\end{aligned}\tag{4.11}$$

Substituting these into the Lagrange Formulation and simplifying yields the dual that depends only on the Lagrange Multipliers α [84].

$$L(\mathbf{w}, b, \alpha) = -\frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n \alpha_i y_i \alpha_j y_j \langle x_i \cdot x_j \rangle + \sum_{i=1}^n \alpha_i \tag{4.12}$$

Then the dual optimization problem is shown in Equation 4.13 and the solution to the dual is the same as the solution to the primal given that the Karush-Kuhn-Tucker conditions (KKT) are satisfied [84]:

$$\begin{aligned}\max_{\alpha_i} & -\frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n \alpha_i y_i \alpha_j y_j \langle x_i \cdot x_j \rangle + \sum_{i=1}^n \alpha_i \\ \text{s.t. } & \alpha_i \geq 0, \quad i = 1, \dots, n \\ & \sum_{i=1}^n \alpha_i y_i = 0\end{aligned}\tag{4.13}$$

The solution to the Lagrangian Multipliers α_i can be found using quadratic programming. It is observed that $\alpha_i > 0$ are called the support vectors and all other $\alpha_i = 0$ [84]. Once α_i is found, \mathbf{w} can be found using the training data and α_i in Equation 4.11. Since the support vectors have been found and known to fall on hyperplane H1 and H2, and the normal \mathbf{w} has been found, the optimal b can be found by isolating for b in H2: $(\mathbf{w} \cdot x_{H2}) + b = 1$ or H1: $(\mathbf{w} \cdot x_{H1}) + b = -1$ for a support vector that falls on H2 or H1 respectively.

Theory of SVM: Not Linearly Separable

The optimization problem for the not linearly separable case is very similar to the linearly separable case, only now there is the addition of a slack variables ζ_i to the constraint (see Figure 4.2). The optimization problem then becomes

(Equation 4.14) [84] with the squared sum of the slack variables modified by a weight C added to the original cost function:

$$\begin{aligned} \min & ||\mathbf{w}||^2 + C \sum_{i=1}^n \zeta_i^2 \\ \text{s.t. } & y_i(\mathbf{w} \cdot \mathbf{x}_i + b) \geq 1 - \zeta_i \forall i \\ & \zeta_i \geq 0 \end{aligned} \tag{4.14}$$

A higher C increases the cost of permitting slack variables and therefore a higher C tends toward perfectly separating the data. The problem is solved in a similar way using Lagrangian Duality and details can be found in Cervantes et al.'s article [84]

Tunable Parameters

In SVM the tunable parameters are C , the type of kernel selected and the kernel's tunable parameter(s). In Section 4.1.2, it was discussed that the larger the C the larger the penalty on slack variables and therefore the classifier tends toward perfect separation of the data points. It doesn't seem like there is agreement on what kernel and what kernel parameter is appropriate for a specific application [84]. However, in practice it is common for datasets with many features to use the basic linear kernel [84]. If the data requires non-linear separation, the Gaussian Kernel is most widely used and its parameter σ can be appropriately chosen using a search [84].

4.1.3 Random Forests

Overview

Random Forests consist of Decision Trees that must be first discussed [86]. A decision tree can be built upon features that are categorical (eg. has astigmatism) or numerical (eg. age). Training a decision tree results in the structure shown in Figure 4.4 with nodes representing the feature and edges representing the options or ranges of values that the feature can take. Typically, the classifier starts at a starting point called the root node. Edges extend uni-directional from the root node presents a decision that the classifier must make based on the features of the new data point. Picking an option or range of values (choosing one of the edges extending from the node) advances the classifier along that edge to the next node in the next level of the tree. The classifier repeats this decision process until there is no more next node; the classifier has reached a "leaf" node and can classify the data point.

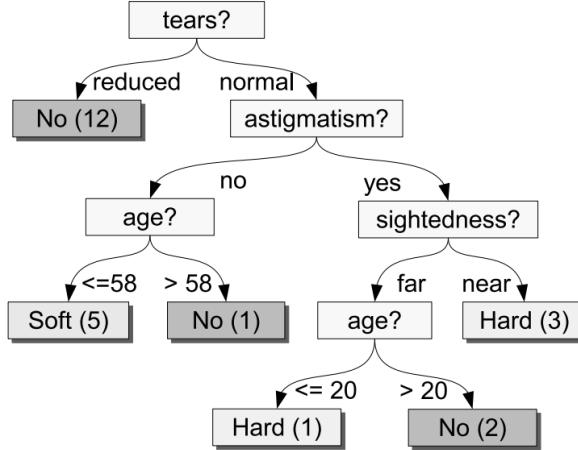


Figure 4.4: Example of a Decision Tree [87]

Random Forests extend the use of Decision Trees by training multiple trees, classifying the new data with each tree and doing a majority vote on the output of each decision tree [88]. Each Decision Tree is trained using a random subset of data drawn from the original training data with replacement. These trees are grown to maximum depth without pruning using the

Classification and Regression Tree (CART) methodology and at each node a random subset of features is used [88]. In situations where there are many features with each feature providing only a small amount of information (eg. medical diagnosis), the single decision tree will have an accuracy slightly better than random classification, whereas using a Random Forest can produce improved accuracy [88].

In time series data, features can be extracted by taking the mean, standard deviation and slope of random subsequences in sequence labelled a certain class 4.5. The number of features is then $3 * \text{the number of subsequences extracted}$. These features are used to train a random forest and new sequences of time series data can be classified by following the same methodology for feature extraction [83].

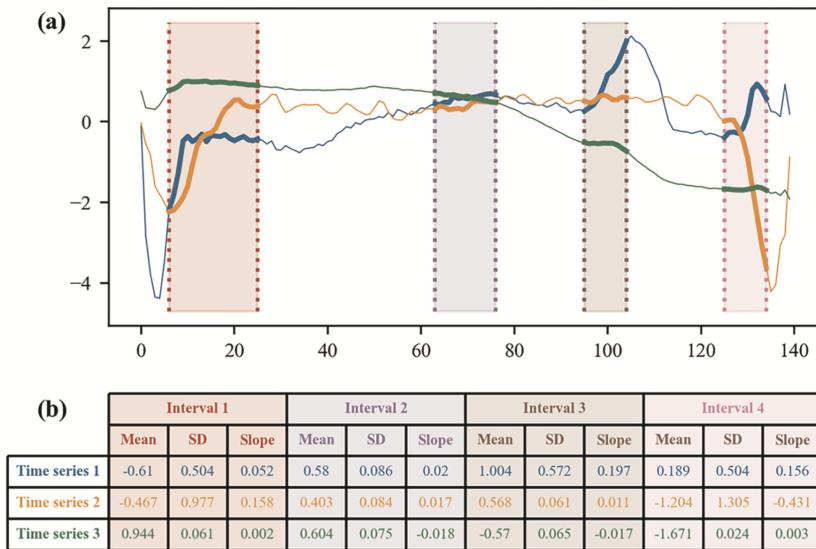


Figure 4.5: Decision Trees used for time-series classification. Mean, standard deviation and slope of random subsequences in a time-series sequence defining a class are used as features. [83]

Classification and Regression Tree (CART) Methodology

The Classification and Regression Tree (CART) method to grow trees outlined by Breiman et al. has the objectivet of splitting the dataset L at each node such that the descendent nodes–nodes one level down–are ”purer” than

the parent node. Purity in sklearn's implementation of decision trees can be evaluated using Gini's Impurity and Entropy [89] and the most common, Gini's Impurity, is shown in Equation 4.15 [89].

$$G = \sum_k p_{mk}(1 - p_{mk}) \quad (4.15)$$

Where p_{mk} is the proportion or probability of observing a class k in subset Q_m

$$p_{mk} = \frac{1}{n_m} \sum_{y \in Q_m} I(y = k) \quad (4.16)$$

At some node m , some subset of the data at node m designated as Q_m , and k classes. At each node, candidate splits $\theta = (j, t_m)$ consisting of a feature j and threshold t_m are evaluated for impurity. According to the source code for sklearn, these candidate splits are chosen by randomly selecting a feature, sorting the feature's values, and using each of the feature's values as a threshold t_m for a candidate split (ie. the thresholds t_m are the values of feature j for the datapoints in the subset Q_m). This process is repeated until the max number of features to be evaluated (an input argument in the Decision Tree Classifier) is reached. The candidate split θ that minimizes the (Gini) impurity is selected to split the data in a binary fashion with 2 descendant nodes [90]. The same process repeats at the descendent nodes recursively until a termination condition (eg. max depth of tree) is reached or the descendant node only contains a single class.

Tunable Parameters

The tunable parameters for Random Forests are similar to Decision Trees and the full list may be found on sklearn's Documentation page for Random Forests [91]. Sklearn mentions that the default values for controlling the size of the tree (eg. `max_depth`, `min_samples_leaf`) leads to large, unpruned and fully grown trees which may be very memory intensive. They recommend to control these parameters to limit the memory consumption [91]. Otherwise, similar to other ML methods, a grid search should be used to find the optimal hyperparameters for the training dataset [92]. Parameters that should be considered for tuning in order of highest positive effect on the Area Under the Curve (AUC) is `mtry` (`max_features` in sklearn), sample size drawn for

training a tree, and the minimum number of observations in a terminal node (min_samples_leaf in sklearn) [92].

4.1.4 Shapelet Transform

Overview

A shapelet is a small but discriminating subsection of a time series waveform that represents a class [93]. To classify with shapelets, the authors in [93] propose using a decision tree with split criteria based on the inclusion shapelets from a shapelet dictionary (inclusion is determined by the distance between a subsequence from a new time-series signal and the shapelet being less than some threshold—Figure 4.6 shows that the inclusion threshold distance is 5.1). Shapelets address some of the shortcomings of using kNN-DTW presented in Section 4.1.1 that result in large time and space complexity and limit its applicability.

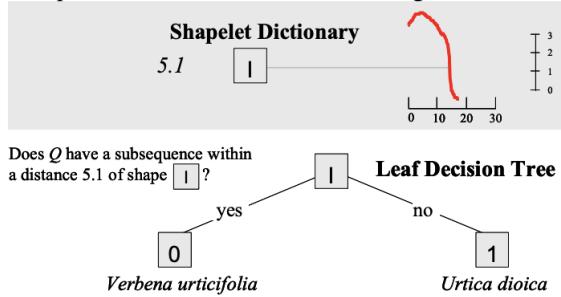


Figure 4.6: Example of a shapelet and the criteria for classifying a new time-series [93].

Determining Shapelets

The authors in [93] mention that the total number of possible shapelets can be calculated by the following equation

$$\sum_{l=MINLEN}^{MAXLEN} \sum_{T_i \in D} (m_i - l + 1) \quad (4.17)$$

for every timeseries T_i in dataset D , l is the length of a candidate shapelet and m_i is the length of the i -th timeseries in the dataset D . For the Trace

dataset with has 200 instances, each with length of 275. If $MINLEN = 3$ and $MAXLEN = 275$, the total number of candidates would be 7,480,200. A brute force approach (Algorithm 1) would be inefficient at finding the best shapelet. The time complexity is $O(\bar{m}^3k^2)$ for average length of timeseries \bar{m} , so the authors propose a pruning strategy to find the best shapelets for classification.

Algorithm 1 Brute Force approach for finding the best shapelet [93], bsf_gain is the information gain calculated using Entropy.

```

candidates  $\leftarrow$  GenerateCandidates( $D, MAXLEN, MINLEN$ )
bsf_gain  $\leftarrow 0$ 
for each  $S$  in candidates do
    gain  $\leftarrow$  CheckCandidates( $D, S$ )
    if gain > bsf_gain then
        bsf_gain  $\leftarrow$  gain
        bsf_shapelet  $\leftarrow S$ 
    end if
end for
return bsf_shapelet

```

To speed up the calculations, the authors use the *early abandon* method in the $CheckCandidates(D, S)$ function. Instead of calculating the entire Euclidean distance between the shapelet and the subsection of the time series, a minimum distance variable (initialized to infinity) is used to track the minimum distance calculated. If, at any point in the Euclidean distance calculation, the minimum distance is exceeded, the calculation is stopped and moves onto the next step. If the full distance between the shapelet and the subsection of the timeseries is less than the minimum distance in memory, then the minimum distance is updated [93].

Another optimization involves *entropy pruning* which occurs once again at the $CheckCandidates(D, S)$ function. Another variable is created to store the best so far information gain. Since the minimum distance calculation between the timeseries and the candidate shapelet is the most costly, these distances are calculated one by one. At each distance calculation, the optimal split point is determined for the distances already calculated and the information gain in the most ideal situation (the rest of Class A on one side and the rest of Class B on the other) is calculated. If the information gain in

the most ideal situation is less than the best so far gain, then the candidate shapelet can be pruned and no further minimum distance calculations will need to be done. Otherwise, continue calculating the minimum distances between the timeseries and the candidate shapelet and checking the information gain for the ideal situation at each step. If after calculating all of the distances between the candidate shapelet and the the timeseries dataset and the information gain is still greater than the best so far information gain, then the best so far gain is updated and the process continues until all of the candidate shapelets have been considered [93].

Multi-Class Classification with Shapelets

To classify multiple classes, the authors of [93] proposed growing a decision tree. At each node a shapelet is found using the optimized Algorithm 1 outlined in the previous subsection. The split point identified using the shapelet at the node splits the dataset into a left and right subset of data. On the left and right subset of data, the optimized Algorithm 1 is used to find the shapelet and the split point for that subset of data. The process continues recursively until a leaf node is reached. This process creates a decision tree and a dictionary of shapelets shown in Figure 4.7 that can be used for classifying time-series data.

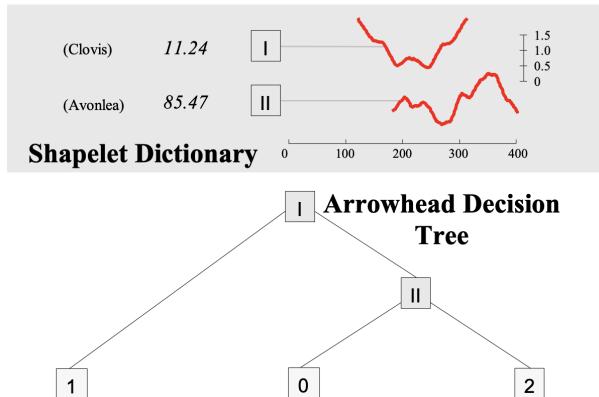


Figure 4.7: An example of a shapelet based decision tree [93].

4.2 Neural Networks

4.2.1 Overview

Neural networks are loosely based on the biological structure of the brain; it consists of nodes called neurons which are connected to each other. Some input x passes through the trained layers of neurons and produces an output $f(x; \theta)$ that can be used in applications such as classification (single point output) and forecasting (vector output) [94]. An example of a neural network is shown in Figure 4.8. Each neuron is a simple function consisting of a weighted sum of the incoming signal added to some bias b , and the result z is passed to an activation function $\sigma(z)$ which determines if the neuron *fires* or not. A very simple example of an activation function called the perceptron is shown in Equation 4.18. Note that due to the loss of information in the perceptron (ie. no information on the strength of activation), however, it is not used in neural networks found in literature [94].

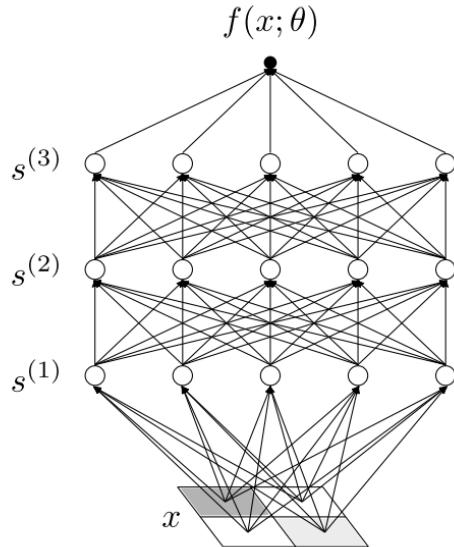


Figure 4.8: A neural network with input x and output $f(x; \theta)$. The input is transformed at each intermediate layer $s^{(1)}, s^{(2)}, s^{(3)}$ until it reaches the output. [94].

$$\sigma(z) = \begin{cases} 1, & z \geq 0 \\ 0, & z < 1 \end{cases} \quad (4.18)$$

There are also different types of layers depending on the type of input x and the application of the neural network. The following subsections will briefly discuss these different types of layers used in their respective neural network. These layers include the linear layers, convolutional layers, recurrent layers, long short term memory (LSTM) layers and transformer layers.

4.2.2 Deep Neural Networks (DNN)

The Neuron

Deep Neural Networks consist of many stacked layers of neurons [94]. The threshold for considering a neural network "deep" is 3 layers [95]. As mentioned in the Overview, a neuron consists of the weighted sum of the incoming signal x added to some bias b , Equation 4.19. The weights for each layer are stored in a weight matrix W .

$$z_i(x) = b_i + \sum_{j=1}^{n_{in}} W_{ij}x_j \text{ for } i = 1, \dots, n_{out} \quad (4.19)$$

Each i -th output goes through an activation function $\sigma_i = \sigma(z_i)$ to determine whether the neuron "fires" to the i -th one in the next layer [94]. Examples of these activation functions can be seen in Figure 4.9.

The inputs and outputs of each layer can then be modeled as follows (Equation 4.20) with superscripts indicating the layer number [94]:

$$\begin{aligned} z_i^{(1)}(x_\alpha) &= b_i^{(1)} + \sum_{j=1}^{n_0} W_{ij}^{(1)} x_{j;\alpha} \text{ for } i = 1, \dots, n_1 \\ z_i^{(l+1)}(x_\alpha) &= b_i^{(l+1)} + \sum_{j=1}^{n_l} W_{ij}^{(l+1)} \sigma(z_j^{(l)}(x_\alpha)) \text{ for } i = 1, \dots, n_{l+1}, l = 1, \dots, L-1 \end{aligned} \quad (4.20)$$

Where L is the total number of layers (the depth of the neural network), and $n_{l=1, \dots, L-1}$ is the number of outputs (the width) at layer l . n_0 and n_L

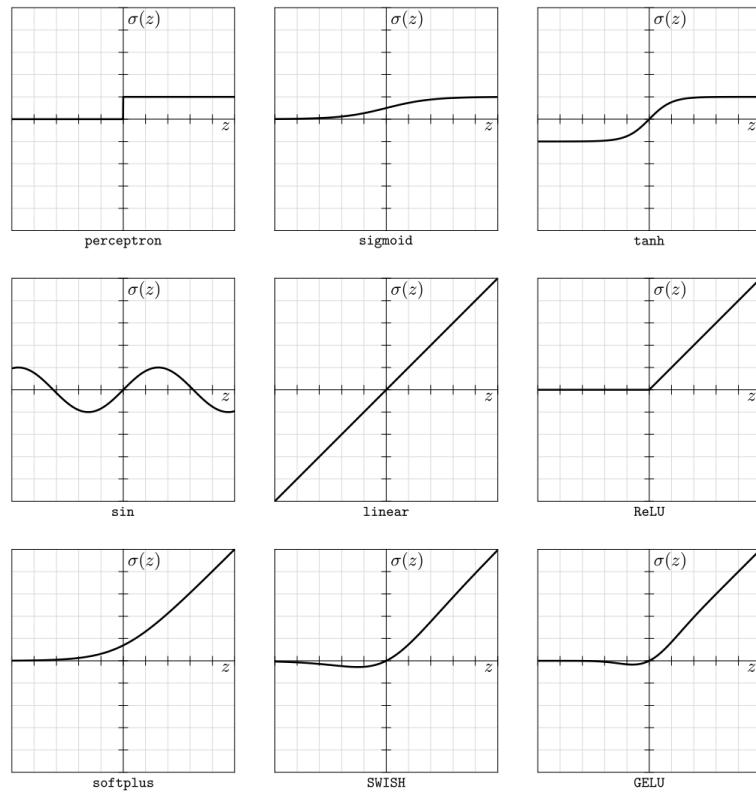


Figure 4.9: Examples of some activation functions [94].

are the input and output dimensions of the neural network respectively [94]. Figure 4.10 depicts the inputs and outputs of a single neuron.

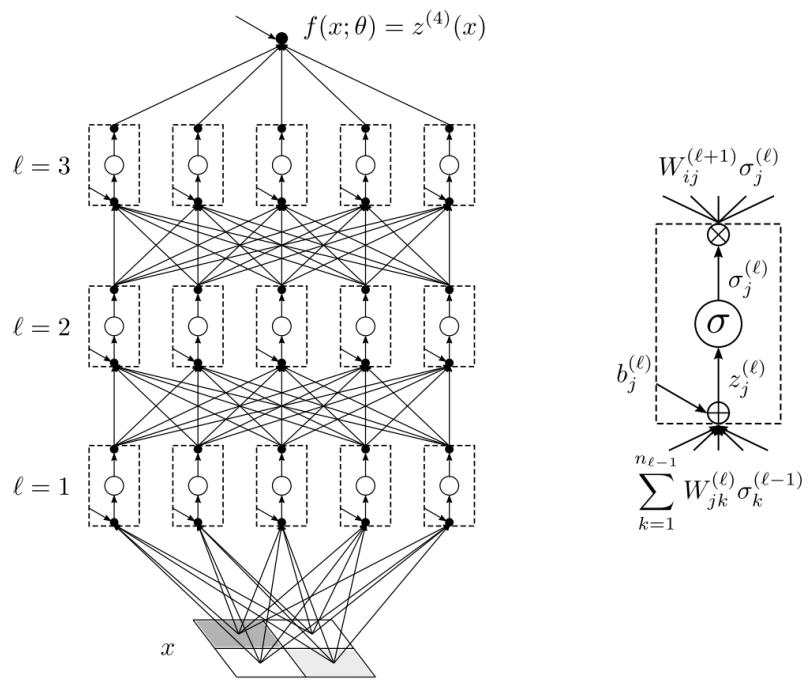


Figure 4.10: Visual representation of the equation of a neuron (Equation 4.20) [94]. The image on the right shows the inputs to the neuron and its corresponding outputs.

Training the Neural Network: Gradient Descent

Training a neural network involves repeatedly updating the weights W_{ij} and biases b_i of the model using a gradient-based method such as gradient descent. Gradient-based methods optimize (minimize) a *loss* function that calculates the error between the output layer result $z^{(L)}(x)$ and the human-annotated labels y ; the goal of the optimization of the loss function is to decrease the error between the output layer result and the labels. Though the choice of loss function depends on the application, an example of a loss function \mathcal{L} can mean-squared error (MSE) shown in Equation 4.21¹ [94].

$$\mathcal{L}_{MSE}(z(x_\delta; \theta), y_\delta) = \frac{1}{2} [z(x_\delta; \theta) - y_\delta]^2 \quad (4.21)$$

With the objective of minimizing the loss in mind, training a simple neural network may have the following steps [94]:

1. Model parameters $W_{ij}^{(l)}$ and $b_i^{(l)}$ (collectively referred to as θ hereon) initialized by independently sampling from an easy to sample distribution such as a zero-mean Gaussian Distribution with variances

$$\begin{aligned} \mathbb{E}[b_{i_1}^{(l)} b_{i_2}^{(l)}] &= \delta_{i_1 i_2} C_b^{(l)} \\ \mathbb{E}[W_{i_1 j_1}^{(l)} W_{i_2 j_2}^{(l)}] &= \delta_{i_1 i_2} \delta_{j_1 j_2} \frac{C_W^{(l)}}{n_l - 1} \end{aligned} \quad (4.22)$$

With variances C and the above presentation uses the Kronecker Delta $\delta_{i_1 i_2}$ (Equation 4.23) to show that the values for each weight and each bias are drawn independently from each other.

$$\delta_{i_1 i_2} = \begin{cases} 1 & \text{if } i_1 = i_2 \\ 0 & \text{if } i_1 \neq i_2 \end{cases} \quad (4.23)$$

2. Using the initial weights and biases, the data from the training dataset \mathcal{A} is passed through the neural network using Equation 4.20 which predicts the last layer $z^{(L)}(x_\alpha; \theta)_{\alpha \in \mathcal{A}}$. The loss is then calculated as the

¹Note that the subscript δ in $(z(x_\delta; \theta), y_\delta)$ is a placeholder to show the δ -th data point in a certain dataset that x and y are from (ie. training dataset \mathcal{A} has data points $(x_\alpha, y_\alpha)_{\alpha \in \mathcal{A}}$ or testing dataset \mathcal{B} has data points $(x_\beta, y_\beta)_{\beta \in \mathcal{B}}$)

average of the loss for each training sample in the training dataset \mathcal{A} and is shown in Equation 4.24

$$\mathcal{L}_{\mathcal{A}}(\theta) \equiv \frac{1}{|\mathcal{A}|} \sum_{\alpha \in \mathcal{A}} \mathcal{L}(z^{(L)}(x_{\alpha}; \theta), y_{\alpha}) \quad (4.24)$$

Also note that there can be multiple neurons in the output layer which the label y_{α} should match in size. The losses from each i -th output neuron with the corresponding i -th portion of the label for the data point α are summed together. In other words:

$$\mathcal{L}_{\mathcal{A}}(\theta) \equiv \frac{1}{|\mathcal{A}|} \sum_{\alpha \in \mathcal{A}} \sum_{i=1}^{n_L} \mathcal{L}(z_i^{(L)}(x_{\alpha}; \theta), y_{i;\alpha}) \quad (4.25)$$

3. The gradient of this loss is calculated with respect to each model parameter θ_{μ} and is used to iteratively update the model parameters in the negative direction of the gradient²

$$\theta_{\mu}(t+1) = \theta_{\mu}(t) - \eta \left. \frac{d\mathcal{L}_{\mathcal{A}}}{d\theta_{\mu}} \right|_{\theta_{\mu}=\theta_{\mu}(t)} \quad (4.26)$$

Where η is a positive hyperparameter called the learning rate and dictates the size of the step in the direction of the negative gradient, μ is added as a subscript to account for all parameters ($[W_{ij}^{(1)}, b_i^{(1)}, \dots, W_{ij}^{(L)}, b_i^{(L)}]$), t is the number of steps in the iterative training process usually starting at $t = 0$.

The derivative $d\mathcal{L}_{\mathcal{A}}/d\theta_{\mu}$ is done iteratively starting at the last layer and propagating the results backwards layer by layer. This step is called

²While Equation 4.26 guarantees at least a local minimum with a small learning rate η . A popular variant of gradient descent called **Stochastic gradient descent** (SGD) uses a random subset of the training data $\mathcal{S}_t \in \mathcal{A}$. \mathcal{S}_t is called a batch and training is organized into "epochs" which is a complete pass through the training dataset. Using SGD has the advantage of better generalization and scalability since training time scales with a fixed size \mathcal{S}_t instead of the entire training set \mathcal{A} (since Equation 4.26 requires the calculation of the gradient for all data points in the data set used, the computation scales linearly with the set used. Assuming that the gradient can be approximated with \mathcal{S}_t , computing the gradient with a smaller subset \mathcal{S}_t is faster than the full \mathcal{A} because there are less gradients to calculate).

backpropagation and using the chain rule, it is shown generalized for any parameter θ_μ in Equation 4.27³:

$$\frac{d\mathcal{L}_A}{d\theta_\mu} = \sum_{i=1}^{n_L} \sum_{\alpha \in \mathcal{A}} \frac{\partial \mathcal{L}_A}{\partial z_{i;\alpha}^{(L)}} \frac{dz_{i;\alpha}^{(L)}}{d\theta_\mu} \quad (4.27)$$

Starting at the last layer L and using the definition of $z_i^{(l+1)}$ in Equation 4.20, parameters associated with the m -th output neuron $m = 1, \dots, n_L$ can be calculated as⁴:

$$\begin{aligned} \frac{d\mathcal{L}_A}{db_m^{(L)}} &= \sum_{i=1}^{n_L} \sum_{\alpha \in \mathcal{A}} \frac{\partial \mathcal{L}_A}{\partial z_{i;\alpha}^{(L)}} \frac{dz_{i;\alpha}^{(L)}}{db_m^{(L)}} = \sum_{\alpha \in \mathcal{A}} \frac{\partial \mathcal{L}_A}{\partial z_{m;\alpha}^{(L)}}(1) \\ \frac{d\mathcal{L}_A}{dW_{mj}^{(L)}} &= \sum_{i=1}^{n_L} \sum_{\alpha \in \mathcal{A}} \frac{\partial \mathcal{L}_A}{\partial z_{i;\alpha}^{(L)}} \frac{dz_{i;\alpha}^{(L)}}{dW_{mj}^{(L)}} = \sum_{\alpha \in \mathcal{A}} \frac{\partial \mathcal{L}_A}{\partial z_{m;\alpha}^{(L)}} \sigma_j^{(L-1)} \end{aligned} \quad (4.28)$$

To get the parameters in the subsequent layer $L - 1$, the chain rule is once again used:

$$\begin{aligned} \frac{d\mathcal{L}_A}{db_j^{(L-1)}} &= \sum_{i=1}^{n_L} \sum_{\alpha \in \mathcal{A}} \frac{\partial \mathcal{L}_A}{\partial z_{i;\alpha}^{(L)}} \frac{dz_{i;\alpha}^{(L)}}{d\sigma_j^{(L-1)}} \frac{d\sigma_j^{(L-1)}}{dz_{j;\alpha}^{(L-1)}} \frac{dz_{j;\alpha}^{(L-1)}}{db_j^{(L-1)}} \\ &= \sum_{i=1}^{n_L} \sum_{\alpha \in \mathcal{A}} \frac{\partial \mathcal{L}_A}{\partial z_{i;\alpha}^{(L)}} W_{ij}^{(L)} \sigma_j'^{(L-1)}(1) \\ \frac{d\mathcal{L}_A}{dW_{jk}^{(L-1)}} &= \sum_{i=1}^{n_L} \sum_{\alpha \in \mathcal{A}} \frac{\partial \mathcal{L}_A}{\partial z_{i;\alpha}^{(L)}} \frac{dz_{i;\alpha}^{(L)}}{d\sigma_j^{(L-1)}} \frac{d\sigma_j^{(L-1)}}{dz_{j;\alpha}^{(L-1)}} \frac{dz_{j;\alpha}^{(L-1)}}{dW_{jk}^{(L-1)}} \\ &= \sum_{i=1}^{n_L} \sum_{\alpha \in \mathcal{A}} \frac{\partial \mathcal{L}_A}{\partial z_{i;\alpha}^{(L)}} W_{ij}^{(L)} \sigma_j'^{(L-1)} \sigma_k^{(L-2)} \end{aligned} \quad (4.29)$$

³Note that the $dz_{i;\alpha}^{(L)}/d\theta_\mu$ term evaluates to zero if the i -th $z_{i;\alpha}^{(L)}$ doesn't depend on the parameter θ_μ .

⁴note that the sum from i to n_L vanishes because $dz_{i;\alpha}^{(L)}/\theta_\mu^{(L)}$ will evaluate to zero if $i \neq m$

More generally, starting from Equation 4.27, the following Equations 4.30 may be used iteratively by sequential multiplication of the chain rule factors to determine the second term in Equation 4.27 (ie. sensitivity of the output in layer $z_{i;\alpha}^{(L)}$ to parameter θ_μ : $dz_{i;\alpha}^{(L)}/d\theta_\mu$) in the l -th layer $l = L - 1, \dots, 1$ (once again working backwards from layer L):

$$\begin{aligned}\frac{dz_{i;\alpha}^{(L)}}{db_j^{(l)}} &= \frac{dz_{i;\alpha}^{(L)}}{dz_{j;\alpha}^{(l)}} \\ \frac{dz_{i;\alpha}^{(L)}}{dW_{jk}^{(l)}} &= \sum_m \frac{dz_{i;\alpha}^{(L)}}{dz_{m;\alpha}^{(l)}} \frac{dz_{m;\alpha}^{(L)}}{dW_{jk}^{(l)}} = \frac{dz_{i;\alpha}^{(L)}}{dz_{j;\alpha}^{(l)}} \sigma_{k;\alpha}^{(l-1)} \\ \frac{dz_{i;\alpha}^{(L)}}{dz_{j;\alpha}^{(l)}} &= \sum_{k=1}^{n_{l+1}} \frac{dz_{i;\alpha}^{(L)}}{dz_{k;\alpha}^{(l+1)}} W_{kj}^{(l+1)} \sigma_{j;\alpha}^{'(l)} \quad \text{for } l < L\end{aligned}\tag{4.30}$$

As a rule of thumb, The change in the loss function \mathcal{L}_A with respect to some parameter $\theta_\mu^{(l)}$ in layer $l < L$ will depend on all of the neurons in the layer above $l + 1$ (causing the summations to stack as the change in the loss function with parameters in lower layers are calculated).

Equation 4.26 is run until there is convergence at a least a local minimum or the number of iterations specified is completed [94].

After the model parameters have been trained, the forward equation (Equation 4.20) can be used on new input data to classify, predict or forecast whatever the model was trained to do.

4.2.3 Convolutional Neural Networks (CNN)

Convolutional Neural Networks (CNN) are typically used for inputs that are in the shape of a 2D arrays with multiple channels (ie. the depth of the input) such as a color image with RBG channels. They are primarily used in image recognition and classification tasks [96]. Unlike the DNN in Section 4.2.2 which requires a set of features to input into the model, CNNs extract the features from images by themselves during the training and do not require the user to determine a set of features. Typically, CNNs are composed of 3 layers [96, 97]:

- Convolutional
- Pooling
- Fully Connected (discussed in 4.2.2)

A typical architecture for a CNN is outlined in Figure 4.11.

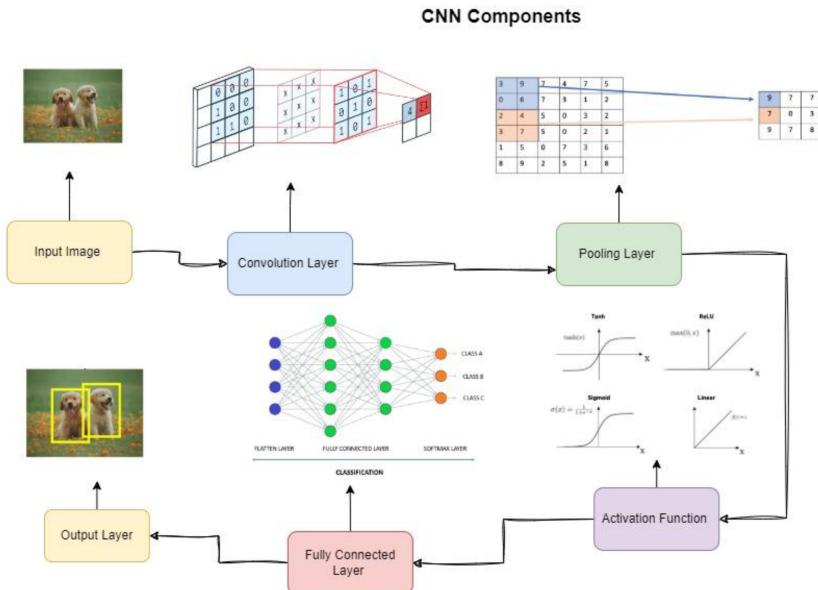


Figure 4.11: Architecture of a common CNN [96].

Convolutional Layer

The convolutional layer involves applying a set of **filters** or **kernels** to the data before it is used (the user can specify the number of kernels k in each layer). These kernels are small 3D arrays whose height and width F are typically smaller than the input array's but have the same depth r as the input array. The kernel's values are called **weights** and are trainable.

Initially, the weights are initialized at random, but as the training progresses, the weights change to enable the kernel to extract meaningful features [96]. In the forward pass, the kernel transforms the image through the dot product between the kernel and the section of the input array. The kernel starts at the top left of the input array and slides left to right, top to bottom calculating the dot product between itself and the section of the input array to produce a single value at each step. How much the filter slides is controlled by the **stride** parameter (ie. a stride of 1 means that the filter will slide to the right 1 element every step whereas a stride of 2 means that the filter will stride to the right 2 elements every step). Figure 4.12 shows the result of applying a 2x2 kernel on a 4x4 input with a stride of 1.

As seen in Figure 4.12 the output height and width was less than the input array's height and width. To avoid the shrinkage in size after applying the kernel, the input array can be zero padded P number of layers. Taye's [96] formula to calculate the expected output dimension's height and width O after applying a kernel with height and width F , input array height and width N , and stride S is shown in Equation 4.31

$$O = 1 + \frac{N + 2P - F}{S} \quad (4.31)$$

The convolutional operation from one kernel will produce an array with shape $(O, O, 1)$. For k number of kernels that the user defines, the final output shape will be (O, O, k) .

After the convolution operation is complete, a bias b is added to the output from each kernel and the values are typically passed through the ReLU activation function σ to remove the negative values⁵ [96, 98]. In other words, the output $z^{k,(l)}$ at layer l for the k -th kernel (that is passed to the next layer) is as follows for some input x [99]:

$$z^{k,(l)} = \sigma(W^{k,(l)} * x + b^k) \quad (4.32)$$

⁵though a Leaky ReLU may be used to avoid the "Dying ReLU" issue

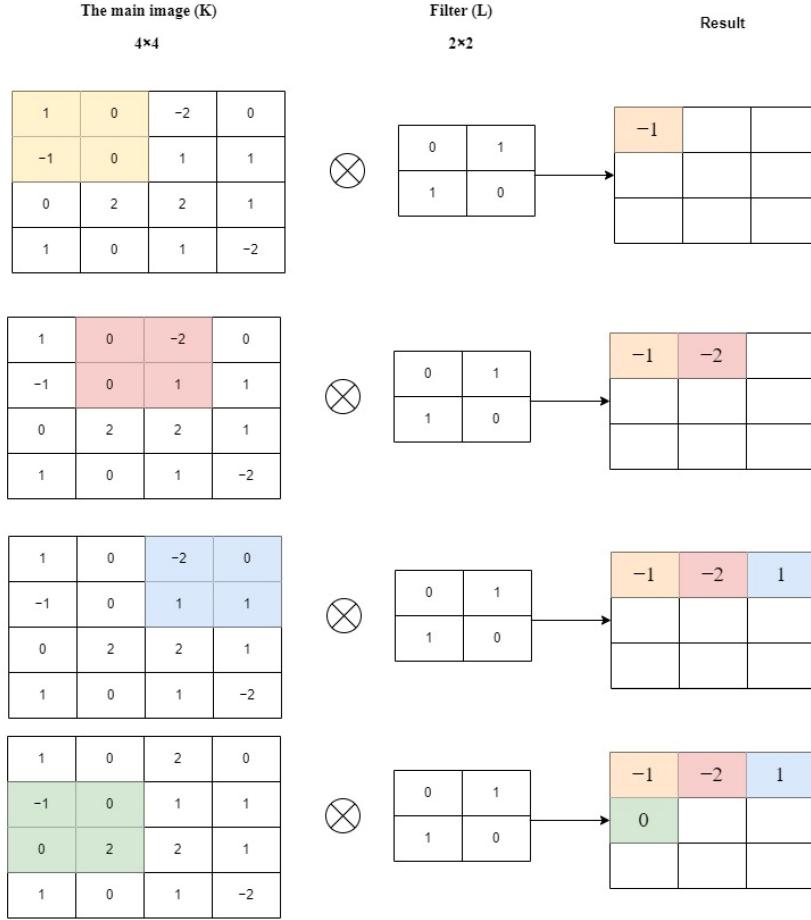


Figure 4.12: Applying a kernel (filter) to the input array [96].

Where $W^{k,(l)}$ are the weights of the k -th kernel, b^k is the bias added to the output of the k -th kernel, and the $*$ symbol represents the convolution operation.

Based on the information discussed on convolutions, given kernel height and width⁶ F , the number of filters k and the depth of the input layer r , the number of trainable parameters in one convolutional layer is shown in Equation 4.33

$$\text{Number of Parameters} = ((F * F * r) + 1)k \quad (4.33)$$

⁶height and width of the kernel do not always need to be the same.

Pooling Layer

Pooling downsamples the output to decrease the height and width dimensionality while retaining the most important data. Similar to the idea of a filter or kernel in the previous section, the pooling filter can have the height and width specified (eg. 2x2) as well as the type (eg. max, min, avg). The pooling filter starts at the top left and slides left to right, top to bottom applying the pooling rule to section of the input to the pooling filter. For example, if the pooling type is a max, the maximum value in the section of the input array where the pooling filter is at is taken and becomes the new value in the shrunken output array. Typically, the pooling filter then slides with a stride equal to its dimensions: $S = F$. The pooling layer operates over all of the channels such that the height and width are reduced, but the depth remains the same [96]. Taye provides an example of Max Pooling and Average Pooling with a 2x2 filter size in Figure 4.13. Similar to the convolutional layer, the output height and width O can be calculated using Equation 4.31 with no padding (ie. $P = 0$) and the depth of the output is the same as the input.

Fully Connected Layer

In CNN architectures such as AlexNet and LeNet-5 [100] and the sample CNN architecture in pyTorch [101] the output from the last pooling layer is flattened into a 1D array and fed as an input to fully connected layers discussed in Section 4.2.2. From the literature, it seems that the ReLU activation function is the most popular for its simplicity and time and resource savings [96, 100, 97].

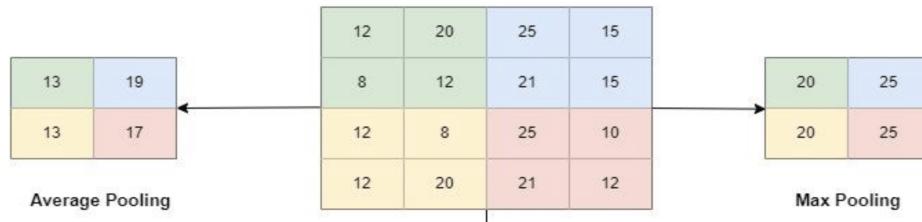


Figure 4.13: The pooling layer [96]

Training the Convolutional Layers

Similar to the training done in the DNN section, training the convolutional layers involve backpropagation and finding the gradient of the cost function with respect to all of the model's parameters. Then using gradient descent, the optimal parameters may be found. Recall Equation 4.26:

$$\theta_\mu(t+1) = \theta_\mu(t) - \eta \frac{d\mathcal{L}_A}{d\theta_\mu} \Big|_{\theta_\mu=\theta_\mu(t)} \quad (4.34)$$

First backpropagation is done through the fully connected layers to find the $d\mathcal{L}_A/d\theta_\mu$ for weights and biases in the fully connected layers. This has already been discussed in Section 4.2.2 through the use of the chain rule. Considering that a typical CNN architecture involves stacking of the Convolutional layers and the Pooling layers prior to the fully connected layers (refer to Figure 4.14 of AlexNet), calculating the weights and biases that are in the Convolutional layers requires backpropagation to work backwards through both the pooling and the convolutional layers. The treatment of the pooling and convolutional layers will be discussed next.

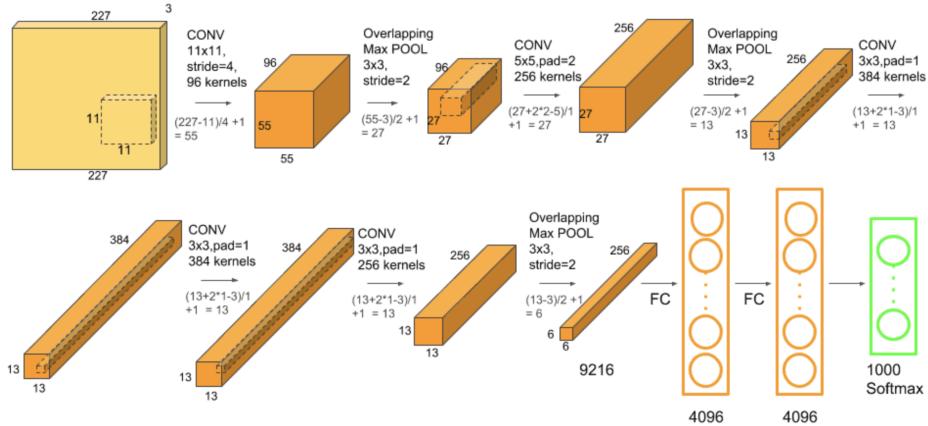


Figure 4.14: Architecture of the AlexNet [102].

Since there are no parameters in the pooling layer, we must look back to the convolutional layer prior to this pooling layer to see how the pooling layer affects the backpropagation. Each cell $z_{ij}^{k,(l)}$ in the output from the convolutional layer is dependent on all the weights in the k -th kernel. Using

ij subscripts to denote the position in the output 2D array, k to denote the k -th kernel, F as the width and height of the kernel, r the depth of the kernel and O the output height and width, Equation 4.32 expanded is shown in Equation 4.35 at layer l :

$$z_{ij}^{k,(l)} = \sigma \left(b^{k,(l)} + \sum_{m=1}^F \sum_{n=1}^F \sum_{d=1}^r W_{mnd}^{k,(l)} x_{[m+i-1][n+j-1][d]} \right) \quad \text{for } i, j = 1, \dots, O \quad (4.35)$$

Since each output cell $z_{ij}^{k,(l)}$ from the k -th kernel depends on each weight in the k -th kernel, to calculate the sensitivity of the loss function to some weight parameter in the k -th kernel of convolutional layer l ($d\mathcal{L}_A/dW_{abc}^{k,(l)}$), the sum of the partial derivatives of all the outputs from the convolution between the k -th kernel and the input with respect to some weight $W_{abc}^{k,(l)}$ must be calculated:

$$\frac{d\mathcal{L}_A}{dW_{abc}^{k,(l)}} = \sum_{i=1}^O \sum_{j=1}^O \frac{\partial \mathcal{L}_A}{\partial z_{ij}^{k,(l)}} \frac{\partial z_{ij}^{k,(l)}}{\partial W_{abc}^{k,(l)}} \quad (4.36)$$

With the equations for sensitivity of loss function to the weights and bias in the convolutional layer, the effect of the Pooling layer can be investigated. As seen in AlexNet (Figure 4.14), there is a MaxPool layer applied to the output of the convolutional layer before being flattened and fed into the fully connected layers. In the forward pass, this pooling layer selects the "winning" cell (in this case the max value in the area selected by the pooling filter) before sliding to the next area. In terms of the output from the convolutional layer in Equation 4.35, only changes in the output $z_{ij}^{k,(l)}$ that have been selected as the "winning" cell by the pooling layer will have an effect on the loss \mathcal{L}_A [103]. Thus, in terms of Equation 4.36, rather calculate all $i, j = 1, \dots, O$, only the set of the winning cells C_{win} (or winning indices) are calculated:

$$\frac{d\mathcal{L}_A}{dW_{abc}^{k,(l)}} = \sum_{ij \in C_{\text{win}}} \frac{\partial \mathcal{L}_A}{\partial z_{ij}^{k,(l)}} \frac{\partial z_{ij}^{k,(l)}}{\partial W_{abc}^{k,(l)}} \quad (4.37)$$

There might also be an Average Pooling layer which has a similar effect to adding a convolutional layer without any additional parameters. In each step, the pooling filter averages the section that it is on:

$$\text{POOL}_{ij}^k = \frac{1}{F \cdot F} \sum_{m=1}^F \sum_{n=1}^F x_{[m+i-1][n+j-1]}^k \quad \text{for } i, j = 1, \dots, O, \quad k = 1, \dots, r \quad (4.38)$$

This pooling function would be added to the chain rule in Equation 4.36 to properly account for the effects of the average pooling layer. For the output dimensions of the average pooling layer O_{AVG} and the output dimensions of the convolutional layer O the equation to find the sensitivity of the loss function to a weight in the convolutional layer is:

$$\frac{d\mathcal{L}_A}{dW_{abc}^{k,(l)}} = \sum_{m=1}^O \sum_{n=1}^O \sum_{i=1}^{O_{AVG}} \sum_{j=1}^{O_{AVG}} \frac{\partial \mathcal{L}_A}{\partial \text{POOL}_{ij}^k} \frac{\partial \text{POOL}_{ij}^k}{\partial z_{mn}^{k,(l)}} \frac{\partial z_{mn}^{k,(l)}}{\partial W_{abc}^{k,(l)}} \quad (4.39)$$

Once again, using the chain rule and Equations 4.36 and 4.37 determined for backpropagating through the convolutional and pooling layers, the sensitivity of the loss function to the remaining parameters can be found.

4.2.4 Recurrent Neural Networks (RNN)

Overview

Long Short Term Memory (LSTM) Neural Networks have impressive benchmarks in literature for tasks including language modeling, speech-to-text transcription, and machine translation [104]. The Recurrent Neural Network (RNN) is a predecessor to and includes as a special case this LSTM neural network [104]. Since LSTM is a type of RNN, RNN will be discussed first to serve as a foundation for LSTM that will be discussed in the next section.

RNNs were developed out of a need for a model that could handle sequential data: data like time-series data, a sentence, data where the value at some time-step t , x_t , depends on the data at the previous time-steps (eg. $x_{t-1}, x_{t-2}, x_{t-3} \dots$). The authors in [105] present the input as a sequence of vectors that each have m features. The sequential input to the RNN can have N time-steps, real or fictitious, $t = 0, \dots, N$. The size of the output hidden state can be specified at some number n . At $t = 0$ the output is calculated very similarly to the neuron Equation 4.19 in section 4.2.2 and passed

through an activation function, in RNN this is typically tanh or ReLU [105]. In vector form with W as an $n \times m$ matrix:⁷

$$h_0(x_0) = \sigma(b + Wx_0) \quad (4.40)$$

What is different from the regular neural network is that h_0 , in addition to being used as a potential output, is used as a hidden state. This hidden state is weighted by the hidden state weight $n \times n$ matrix U and used to calculate the output hidden state of the next time step:

$$h_1(x_1) = \sigma(Uh_0 + Wx_1 + b) \quad (4.41)$$

More generally [105]⁸:

$$h_t(x_t) = \sigma(Uh_{t-1} + Wx_t + b), \quad t = 0, \dots, N \quad (4.42)$$

For hidden states used as an output, they are passed through a linear layer which results in the output y_t vector with a size of r .

$$y_t(h_t) = Vh_t + c, \quad t = 0, \dots, N \quad (4.43)$$

Where V is another weight matrix with dimensions $r \times n$ and c is the bias with a size of r .

Note that the weight matrices U , W , and V in the RNN are reused in each time step for all $t = 0, \dots, N$. In other words, the number of parameters will not grow as the length of the sequence increases or N grows.

A visual depiction of the architecture of the RNN is shown in Figure 4.15 [106].

Training the Simple Recurrent Neural Network

Once again training the recurrent neural network involves the idea of back-propagation. For RNNs, in addition to backpropagating through the layers, there will need to be back propagation through time (BPTT) [105]. The authors in [105] provide equations for the calculation of the RNN parameters:

⁷note that the subscript now indicate a time-step

⁸Note that typically $h_{-1} = 0$ as there is no negative time

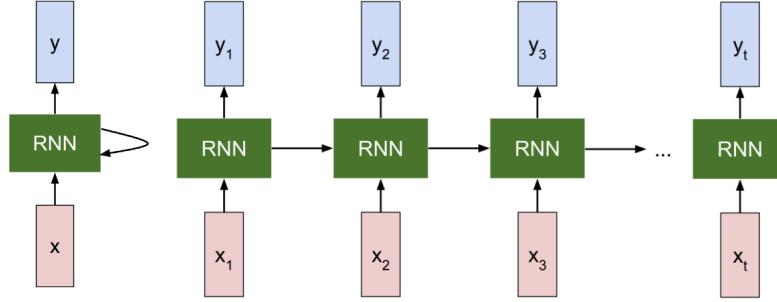


Figure 4.15: A visual depiction of an RNN. The far left shows the simplified RNN with x being passed in, the middle looping arrow represents the calculation of the hidden state at each time step and being passed to the calculation for the hidden state in the next time step. Each hidden state can be used to calculate an output y using Equation 4.43. The connected structure to the right of the simplified RNN shows the "unrolled" version of the RNN. The arrow to the right shows the hidden state being passed onto the calculation in the subsequent time step [106].

$$\begin{aligned} \frac{\partial \mathcal{L}_{\mathcal{A}}}{\partial V} &= \sum_{t=0}^N \left(\frac{\partial \mathcal{L}_{\mathcal{A}}^t}{\partial y_t} \right) \frac{\partial y_t}{\partial V} = \sum_{t=0}^N \left(\frac{\partial \mathcal{L}_{\mathcal{A}}^t}{\partial y_t} \right) h_t^T \\ \left(\frac{\partial \mathcal{L}_{\mathcal{A}}}{\partial c} \right)^T &= \sum_{t=0}^N \left(\frac{\partial \mathcal{L}_{\mathcal{A}}^t}{\partial y_t} \right)^T \frac{\partial y_t}{\partial c} = \sum_{t=0}^N \left(\frac{\partial \mathcal{L}_{\mathcal{A}}^t}{\partial y_t} \right)^T (1) \end{aligned} \quad (4.44)$$

The partial derivatives for the bias c is transposed to retain the chain-rule derivative for scalar [105].

For the gradients of the parameter U , W and b , the partial derivative of the loss with respect to the intermediate function in Equation 4.42 ($z_t = Uh_{t-1} + Wx_t + b$) will be defined. Note that to find the gradient of the parameters, the algorithm is required to step backwards in time starting from the $t = N$ and iteratively using Equation 4.45.

$$\begin{aligned} \lambda_t &= \frac{\partial \mathcal{L}_{\mathcal{A}}}{\partial z_t} \\ &= (U\sigma'_t)^T \lambda_{t+1} + (V\sigma'_t)^T \frac{\partial \mathcal{L}_{\mathcal{A}}}{\partial y_t}, \quad 0 < t \leq N - 1 \end{aligned} \quad (4.45)$$

With the intermediate partial derivative λ_t defined the gradients of the other parameters are as follows:

$$\begin{aligned}\frac{\partial \mathcal{L}_{\mathcal{A}}}{\partial U} &= \sum_{t=0}^N \left(\frac{\partial \mathcal{L}_{\mathcal{A}}^t}{\partial z_t} \right) \frac{\partial z_t}{\partial U} = \sum_{t=0}^N \lambda_t h_{t-1}^T \\ \frac{\partial \mathcal{L}_{\mathcal{A}}}{\partial W} &= \sum_{t=0}^N \left(\frac{\partial \mathcal{L}_{\mathcal{A}}^t}{\partial z_t} \right) \frac{\partial z_t}{\partial W} = \sum_{t=0}^N \lambda_t x_t^T \\ \left(\frac{\partial \mathcal{L}_{\mathcal{A}}}{\partial b} \right)^T &= \sum_{t=0}^N \left(\frac{\partial \mathcal{L}_{\mathcal{A}}^t}{\partial z_t} \right)^T \frac{\partial z_t}{\partial b} = \sum_{t=0}^N (\lambda_t)^T\end{aligned}\quad (4.46)$$

From inspecting the equation for λ_t , it can be seen that gradient of the parameter matrix U and W will require U and V to be nested multiple times. For example with expansion to the next time step:

$$\lambda_t = (U\sigma'_t)^T \left[(U\sigma'_{t+1})^T \lambda_{t+2} + (V\sigma'_{t+1})^T \frac{\partial \mathcal{L}_{\mathcal{A}}}{\partial y_{t+1}} \right] + (V\sigma'_t)^T \frac{\partial \mathcal{L}_{\mathcal{A}}}{\partial y_t} \quad (4.47)$$

The nesting of the parameter matrices in the calculation of λ_t causes small changes in the these parameter matrices W and U to be greatly amplified as the sequence of data gets longer leading to the vanishing gradient or exploding gradient issues that typically make these simple RNNs unsuitable for practical use [105]. Although workarounds such as clipping or setting a threshold on the upper and lower bound of the parameter values seems to alleviate some of these issues, an approach more widely used are LSTMs discussed in the next section [105].

4.2.5 Long Short Term Memory (LSTM) Neural Networks

Long Short Term Memory (LSTM) Neural Networks build upon idea of the simple RNN and allow for information for the sequence of data to be learned while mitigating the issue of the vanishing gradient and the exploding gradient. The LSTM adds a **memory cell** c_t to the simple RNN architecture. In comparing with the simple RNN, the original calculation for h_t in the simple RNN becomes an intermediate output \tilde{c}_t in the LSTM that is combined with

the memory cell before being output as a hidden state to be used in the calculation for subsequent time steps. The equations for the LSTM are as follows [105]:

$$\begin{aligned}\tilde{c}_t &= g(U_c h_{t-1} + W_c x_t + b) \\ c_t &= f_t \odot c_{t-1} + i_t \odot \tilde{c}_t \\ h_t &= o_t \odot g(c_t)\end{aligned}\tag{4.48}$$

With \odot representing element-wise multiplication and i_t , f_t and o_t representing the *input*, *forget*, *output* gates respectively. The equations for these gates are as follows:

$$\begin{aligned}i_t &= \sigma(U_i h_{t-1} + W_i x_t + b_i) \\ f_t &= \sigma(U_f h_{t-1} + W_f x_t + b_f) \\ o_t &= \sigma(U_o h_{t-1} + W_o x_t + b_o)\end{aligned}\tag{4.49}$$

With the following variables

- g : an activation function such as logistic function, tanh, or ReLU. Typically tanh [105]
- σ : an activation function such as logistic function, tanh, or ReLU. Typically Logistic to facilitate the "gating" function of forcing the output of the gate to be between 0 and 1 [105].
- U_{step} : The U parameter matrix for *step* in the LSTM. Values of the $step \in \{c, i, f, o\}$ for the *memory cell*, *input*, *forget* and *output* steps respectively
- W_{step} : The W parameter matrix for the *step* in the LSTM. Values of the $step \in \{c, i, f, o\}$ for the *memory cell*, *input*, *forget* and *output* steps respectively
- b_{step} : The b bias parameter for the *step* in the LSTM. Values of the $step \in \{c, i, f, o\}$ for the *memory cell*, *input*, *forget* and *output* steps respectively

Similar to the RNN the parameters in the memory cell, input, forget and output steps are reused at calculation for each time step so the number of parameters do not grow as the sequence gets longer. The total number of parameters for the LSTM is 4 times that of a RNN: $n_{\text{LSTM, params}} = 4(n^2 + nm + n)$ (recall that n is the number of hidden state features and m is the input vector number of features).

The architecture of the LSTM unit is shown in Figure 4.16

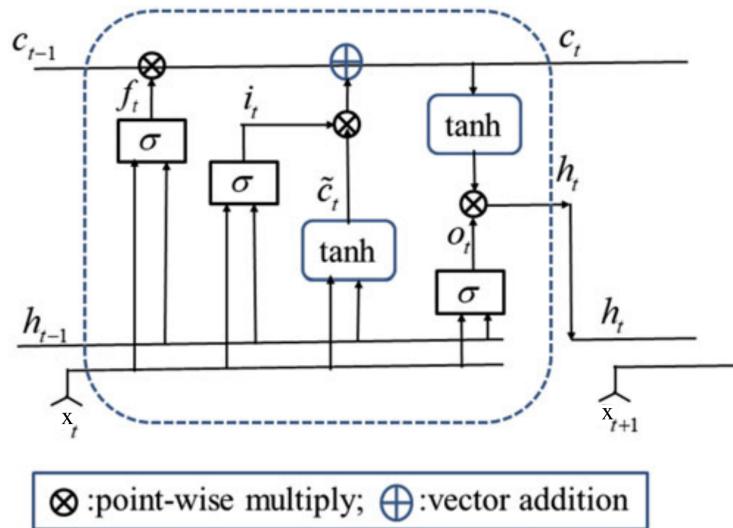


Figure 4.16: LSTM Architecture [105]. Note that the activation function g is shown as the tanh function.

Training LSTMs

The author in [105] does not provide equations for the backpropagation of the LSTM. However, since the LSTM is a type of RNN, the backpropagation through time (BPTT) would be used to find the gradients of the parameters $U_{step}, W_{step}, b_{step}$ for all of the steps $\{c, i, f, o\}$ by working backwards from the loss function $\mathcal{L}_{\mathcal{A}}$ to the point when these parameters are used.

4.2.6 Transformer Neural Networks

One limitation of the simple RNN and the LSTM is the sequential nature of the calculations. According to Vaswani et al. the sequential nature of the calculation (ie. h_t depends on the calculation of h_{t-1}) prevents any form of parallelization to occur limiting training speed and memory requirements increase with length of the sequences [107].

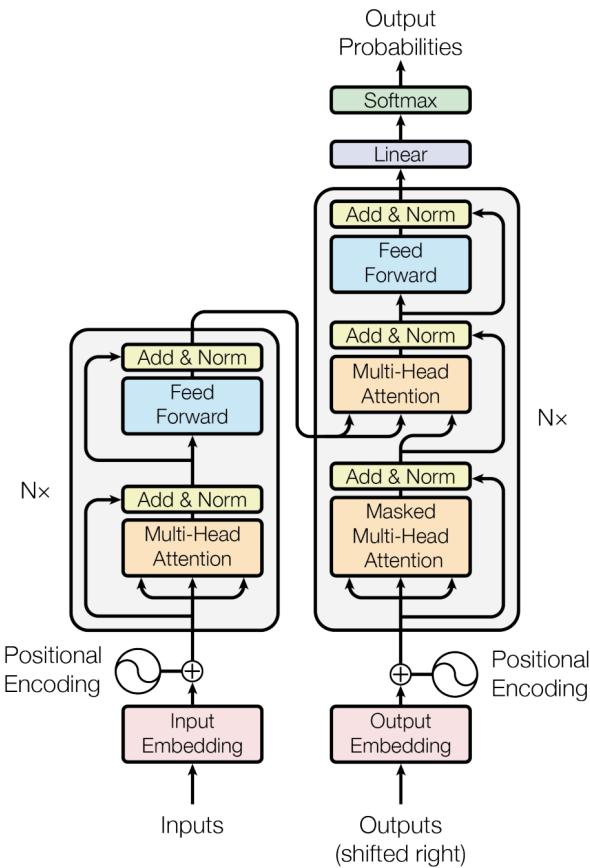


Figure 4.17: Transformer Architecture [107]

The transformer architecture that is proposed in [107] eliminates the sequential dependence and uses only the attention mechanism to draw global dependencies between the input and the output [107]. Although Vaswani et al. proposed the architecture for the task of machine translation (and more

recently natural language processing), the idea of leveraging contextual information in a sequence may be useful in the classification of the time series data in this thesis. The transformer architecture is shown in Figure 4.17.

Positional Encoding

The first step in the transformer architecture (Figure 4.17) is the positional encoding step. An input matrix X of d_{model} features and N time steps⁹ (ie. $X \in \mathbb{R}^{N \times d_{model}}$) needs to have some sort of positional information injected to make use of the sequence since the transformer does not use recurrence or convolution [107]. Sine and Cosine functions were used to positionally encode each vector in X [107]:

$$\begin{aligned} PE_{(pos,2i)} &= \sin(pos/10000^{2i/d_{model}}) \\ PE_{(pos,2i+1)} &= \cos(pos/10000^{2i/d_{model}}) \end{aligned} \quad (4.50)$$

Where i represents the i -th dimension or feature, and pos represents the time-step or position. The $PE \in \mathbb{R}^{N \times d_{model}}$ matrix is added to X to positionally encode the input matrix.

Attention

After positionally encoding the input matrix, Vaswani et al. use a form of the attention mechanism as a "Scaled Dot-Product Attention" in [107]. In general, the input to the attention function consists of queries and keys of dimension d_k and values with dimensions d_v . The queries and keys are dot-product together, divided by $\sqrt{d_k}$, and put through a softmax to obtain the weights on the value. Queries, Keys and Values are compactly represented in the form $Q \in \mathbb{R}^{N \times d_k}$, $K \in \mathbb{R}^{N \times d_k}$, $V \in \mathbb{R}^{N \times d_v}$ respectively¹⁰ and are used as follows in Equation¹¹ 4.51

⁹Note that in natural language processing (NLP), these time steps are tokens–words or special characters depending on the tokenization function [108]

¹⁰Note that N used here refers to the number of time steps, where as in [107] N refers to the number of stacked Attention + Feed Forward layers

¹¹note that there is a "masked" version of this equation. Masking ensures that positions don't attend to subsequent positions (since early positions shouldn't have knowledge of future positions). As seen in Figure 4.18, masking is applied after the scaling step and before softmax in the scaled Dot-product attention. for some query vector q_j at position j and key vector k_l at position l , if $l > j$ then $q_j \cdot k_l = -\infty$

$$\text{Attention}(Q, K, V) = \text{softmax} \left(\frac{QK^T}{\sqrt{d_k}} \right) V \quad (4.51)$$

A visual diagram of the Scaled Dot-Product Attention is shown in Figure 4.18.

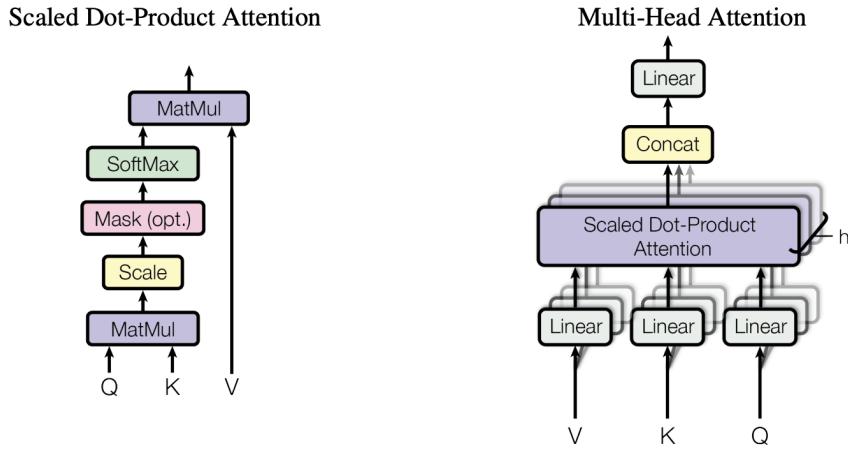


Figure 4.18: Attention. (left) Scaled-Dot Product Attention, (right) Multi-head attention. [107]

Computing Equation 4.51 once is referred to a single head of attention [107]. For a model with an input dimension of d_{model} Vaswani et al. employ a multi-headed attention in their transformer architecture by linearly projecting the queries, keys and values to a lower dimension d_k , d_k and d_v respectively. Each i head of attention has an associated trainable query weights W_i^Q , key weights W_i^K , and value weights W_i^V . Given a model input dimension d_{model} and total number of heads h , the lower dimensions d_k and d_v are calculated as follows: $(d_k = d_v = d_{model}/h)$.

For example, in [107], the input to the transformer are embeddings (a numerical representation of words that encode semantic and contextual information [109]) with 512 dimensions. Using 8 heads $h = 8$, the dimensions of the query, key and value for each head is $d_k = d_v = 512/8 = 64$ [107]. The resulting values from the calculation of each head of attention are concatenated together and matrix multiplied with an output weight matrix W^O .

The equation for multi-head attention in [107] is modified for some input matrix $X \in \mathbb{R}^{N \times d_{model}}$ and shown in Equation 4.52

$$\begin{aligned} \text{MultiHead}(Q, K, V) &= \text{Concat}(\text{head}_1, \dots, \text{head}_h)W^O \\ \text{where } \text{head}_i &= \text{Attention}(XW_i^Q, XW_i^K, XW_i^V) \end{aligned} \quad (4.52)$$

Where $W_i^Q \in \mathbb{R}^{d_{model} \times d_k}$, $W_i^K \in \mathbb{R}^{d_{model} \times d_k}$, $W_i^V \in \mathbb{R}^{d_{model} \times d_v}$, and $W^O \in \mathbb{R}^{hd_v \times d_{model}}$ [107]. These weight matrices are all learned in practice [107].

The output of the multi-head attention block is a matrix that is the same shape as the input matrix X and contains information on how the surrounding context should affect the values in each embedding vector. The output from the multi-head attention and X are added together and normalized before being passed onto the next block: the Fully Connected layer.

Aside: Attention Learning Intuition in NLP

Based on Equation 4.52, each embedding in X gets converted into a query q , key k and value v row vector using the respective weight matrices. These row vectors are vertically stacked forming Q, K and V . In the step with the matrix multiplication between QK^T in Equation 4.51, each query is dot-product with each key resulting in a $N \times N$ matrix that shows how similar the j -th query is to the l -th key.

$$QK^T = \begin{bmatrix} q_1 \cdot k_1 & \dots & q_1 \cdot k_N \\ \vdots & \ddots & \vdots \\ q_N \cdot k_1 & \dots & q_N \cdot k_N \end{bmatrix} \quad (4.53)$$

In consideration of the QK^T step, the weight matrices for the query and key (W_i^Q and W_i^K respectively) are trained so that if there is a relationship between a query at j : q_j and key at l : k_l , then the vectors q_j and k_l will be similar (ie. produce relatively large positive value when dot-product). The QK^T step is responsible determining which elements in the sequence should affect which other elements in the sequence and the magnitude of the effect.

Since the "what" is handled by the QK^T , the "how" is handled by V and W^O . Through training, W_i^V and W^O together should learn the direction (in terms of vectors) that the l -th embedding (from the key) should add to

the j -th embedding (from the query) that had relationships discovered in the QK^T step. Since

$$V = \begin{bmatrix} v_1 \\ \vdots \\ v_N \end{bmatrix} \quad (4.54)$$

And treating the softmax and division by $\sqrt{d_k}$ as a function $f()$, the output of a single head of attention is shown in Equation 4.55¹²

$$f(QK^T)V = \begin{bmatrix} f(q_1 \cdot k_1) * v_1 + \dots + f(q_1 \cdot k_N) * v_N \\ \vdots \\ f(q_N \cdot k_1) * v_1 + \dots + f(q_N \cdot k_N) * v_N \end{bmatrix} \quad (4.55)$$

All the heads of attention are concatenated together and matrix multiplied by W^O to project the heads of attention from d_v back to d_{model} (meaning W^O has a part in influencing the "direction" as well).

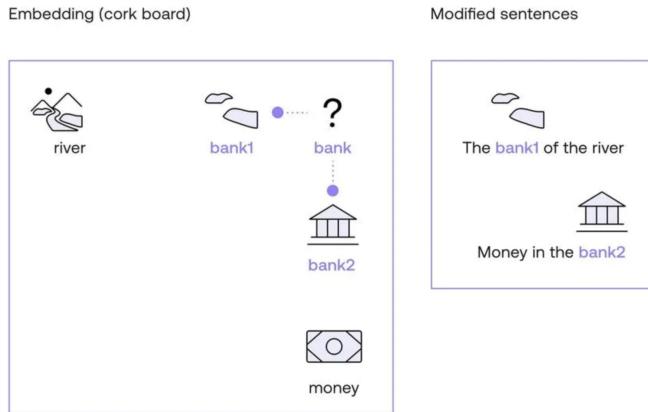


Figure 4.19: The river bank pushes the embedding of bank to the left closer to the river related words. Money pushes "bank" embedding toward the financial related words [110].

¹²considering how each row is an embedding, notice how the value and key subscripts increment in tandem. This can be interpreted as the embedding of key's effect on the embedding of the query.

In terms of NLP, the output of the multi-head attention nudges the original embeddings in X in the direction that is more semantically appropriate based on the context. [110] uses the example of the word "bank". Based on the context "bank" could be either a "the bank of the river" or "money in the bank". Keeping in mind Equation 4.53, well-trained query weights and key weights should produce similar vectors when "bank" is the query and "river" or "money" is the key. With value vectors defining a direction that's added to the original embeddings, in this example, "river's" value vector should nudge the "bank's" embedding closer to "river" related embeddings and "money's" value vector should nudge "bank's" embedding closer to the finance related embeddings, Figure 4.19 [110].

Feed Forward (Fully Connected)

After the input is modified by the output of the multi-head attention layer. A fully connected layer is applied to each position separately and identically with a ReLU activation in between [107].

$$\text{FFN}(x) = \max(0, xW_1 + b_1)W_2 + b_2 \quad (4.56)$$

The output from the feed-forward fully connected layer is then added and normalized with the output from the previous layer as shown in the transformer architecture Figure 4.17. The feed forward and attention layers are repeated as many times as the user specifies and eventually outputs the probabilities (prediction).

Training a Transformer

The authors in [107] do not go into the mathematical theory for training the transformer architecture. It is assumed that similar to the other architectures, backpropagation is used to train the weights W^O , and each of the W_i^Q , W_i^K and W_i^V in the multi-headed attention step. Liang in [111] uses calculus to find the gradients of these parameter matrices and has also validated it with pyTorch's implementation of the MultiHeadAttention.

4.3 Classification Strategy

The objective of the thesis is to use positional + IMU sensor data to create a system that can classify fine-grained cooking tasks in real-time. Once again, this real-time classification system would enable ADL quality monitoring which would allow for the early detection of physical or cognitive decline [61].

Model and Input

Time series data is a series of data that each consists of a value at an associated timestamp. In terms of the system investigated in this thesis, this could be a series of position in the x-axis measured at time $t_1, t_2, t_3\dots$ respectively. The data can be said to be indexed by time [112] typically in ascending order.

Table 4.1: Summary of the models discussed in this section along with their Input. Acronyms used: Feature Extraction (FE), Time-series (TS)

Model	FE?	Input
kNN	Yes	Extracted features from a subsequence
kNN with DTW	No	The full subsequence
SVM	Yes	Extracted features from a subsequence
Random Forests	Yes	Extracted features from a subsequence
Shapelet Transform	No	The full subsequence
DNN with Features	Yes	Extracted features from a subsequence
DNN with TS	No	TS with constant $n_{samples}$
1D CNN [113]	No	TS with constant $n_{samples}$
LSTM	No	TS with variable $n_{samples}$ [114]
Transformer NN	No	TS with variable $n_{samples}$

Based on the models investigated in this section, some can take the time-series input as is, and some require "Feature Extraction" or compressing the data down into a feature vector that can be used with the model. In the section for Random Forests, random subsequences of a time-series (with

a minimum length) was taken and the subsequence was compressed into a feature vector consisting of the Mean, Standard Deviation and the Slope [83]. The Time-series Feature extraction library (TSFEL) is a library that Barandas et al. built for the time-series feature extraction task and has over 60 features across temporal, statistical and spectral domains [115]. In addition to the mean, standard deviation, and slope, features can include the min, max, median, absolute energy, max frequency, median frequency and many others [116]. Table 4.1 summarizes the models discussed and the type of input required.

Real-Time Considerations

In addition to the classification model that must be evaluated, the system's practicality in real-time application must be investigated. The following are considerations for real-time classification:

- The latency of the model or the speed of the classification. The latency of classification must be lower than the time taken to collect the data for classification. If the latency of classification is higher than the data collection interval, then the classification system will not be able to keep up with the data input.
- Robustness to temporal variation. The task speed is expected to vary greatly depending on multiple factors such as skill and health conditions. For example, able-bodied adults would be expected to complete tasks faster than older adults with comorbidities.
- Real-time activity boundary detection. Since each task inherently varies in time, for example opening a door versus continuous chopping, the system must be able to segment out each task within the stream of data.

Prior to addressing these concerns, the characteristics of the data must be first investigated. The next section will detail the selection of cooking tasks, the task breakdown and the resulting data collected from performing the experiments.

Chapter 5

Pilot Testing of Detecting Activities to make a Sandwich

Continuing from the findings in Chapter 3, the 9H anchor configuration was used to perform a preliminary classification of activities in the kitchen. Steps performed in making a sandwich were broken down and organized into Setup, Preparation, Cooking, and Finishing steps, Figure 5.1.

From the actions shown in Figure 5.1, actions with distinct location or patterns were selected as classes for classification. OPENFRIDGE, OPENFREEZER, and GETPLATE were selected as classes from the Setup category, washing hands/vegetables/fruits/using the kitchen sink were grouped into a WASHHANDS category, and SLICETOMATO were selected as a class. Finally, All intermediary transitions or motionless segments were grouped into a UNDEFINED category. Single trials were performed to collect data for each of these classes.

Assumption

Making a Club Sandwich
(turkey, bacon, cheese, lettuce, tomato)

<https://www.foodnetwork.com/recipes/food-network-kitchen/classic-club-sandwich-recipe-2117730>

- Making sandwich will be at the table,

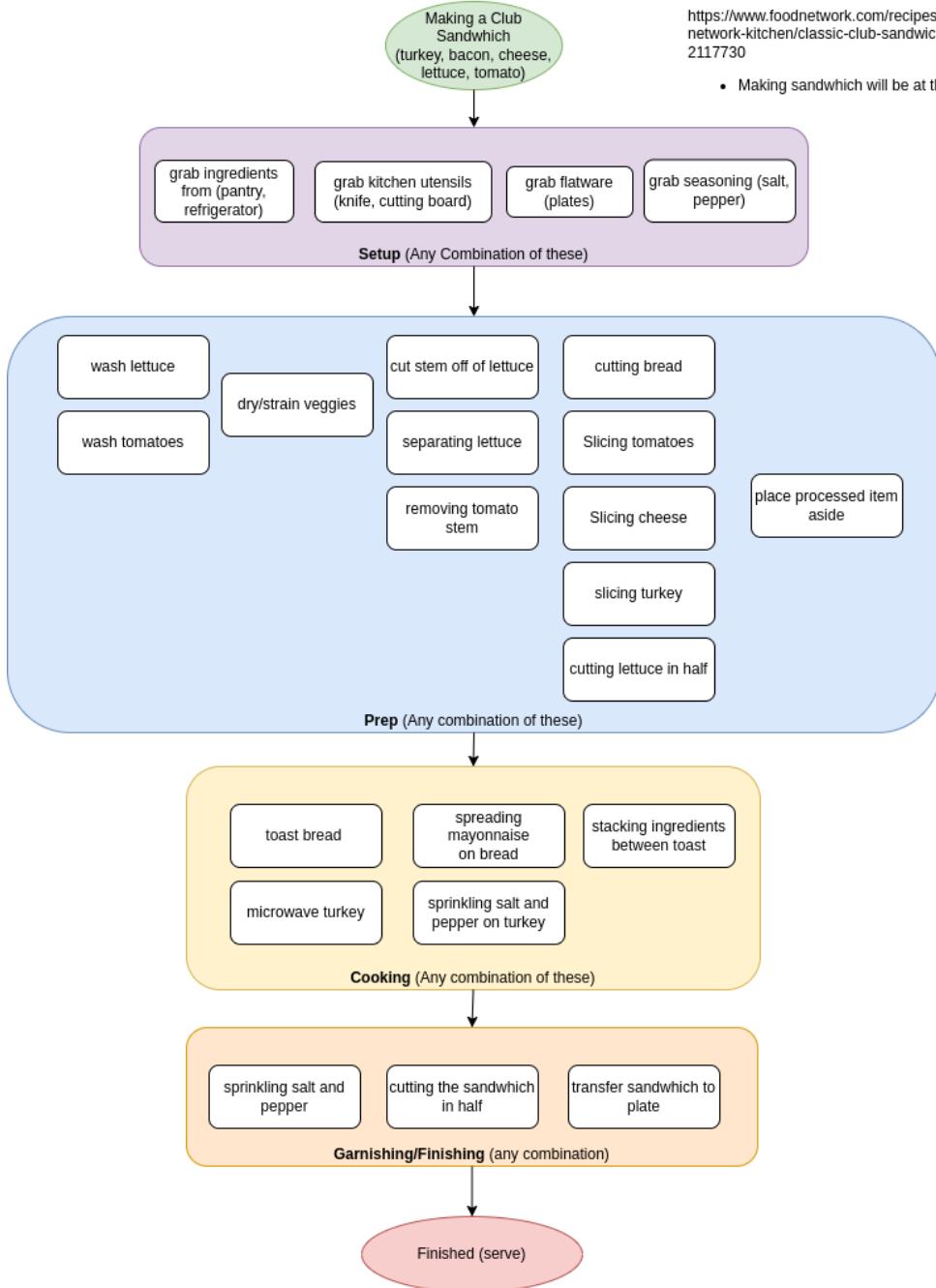


Figure 5.1: Task decomposition of making a sandwich.

5.1 Experimental Protocol

5.1.1 Setup

Several points were enforced to ensure that the training dataset captures the variation in action sufficiently when classifying data from right-handed individuals.

- Pozyx Tag is mounted on the right wrist (Figure 5.2).
- Initial position for each of the single trials are not marked. Participant will be able to choose a location from which they can perform the action comfortably without moving their feet.
- An action starts when the individual contacts the appliance or furniture. For SLICETOMATO the action starts when an individual starts slicing the tomato and ends when they stop slicing the tomato. Motions such as picking up the knife and getting in position to slice were considered transitions and labelled as UNDEFINED.

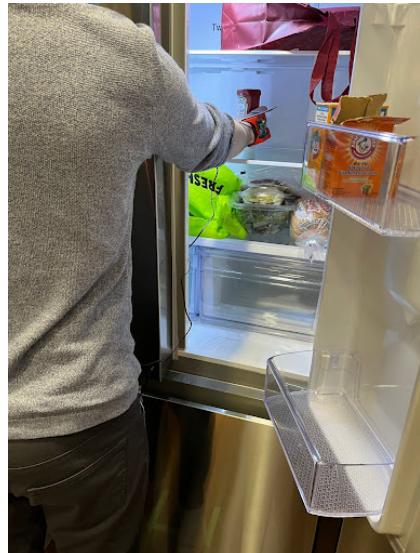


Figure 5.2: Pozyx tag mounted on the wrist. The participant is performing the OPENFRIDGE task

5.1.2 Data Collection

Custom Python stopwatch scripts were created to accurately label periods of transitions (quiet standing + getting into position for the action) and the action. An example of the data collected is shown in Figure 5.3. For each action there is a quiet standing period at the beginning and end. OPENFRIDGE, OPENFREEZER, OPENPLATE, WASHHANDS each had 5 repetitions for each trial. SLICETOMATO contained 3 slices to conserve the amount of tomato. Each action had a total of 5 trials.

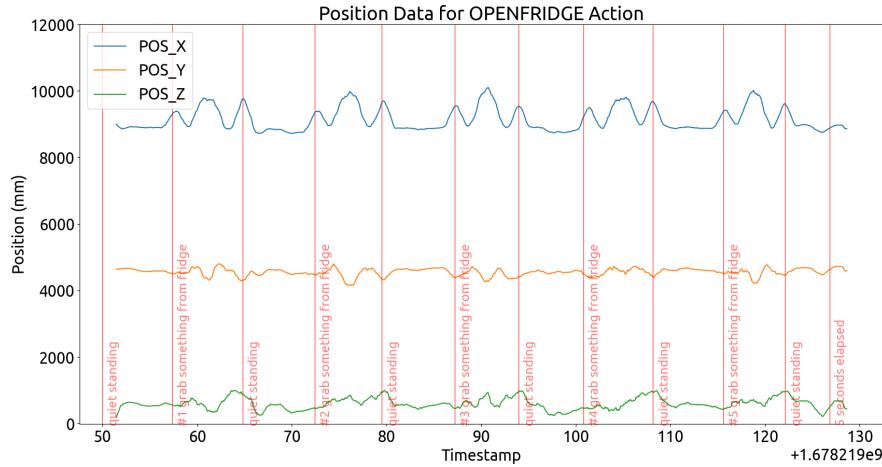


Figure 5.3: Labelled position data of the OPENFRIDGE action. Note that the "quiet standing" periods do not consist entirely of quiet standing, but also include traces of transitions from getting into the correct position to perform the action.

Since the Pozyx Tag contained a BNO055 chip, in addition to 3D Position data, the tags were able to capture inertial data including Accelerometer Data, Linear Accelerometer Data, Angular Velocity Data, and the orientation.

Data for each of the actions that relate to making a sandwich were collected from 2 participants.

5.2 Feature Extraction

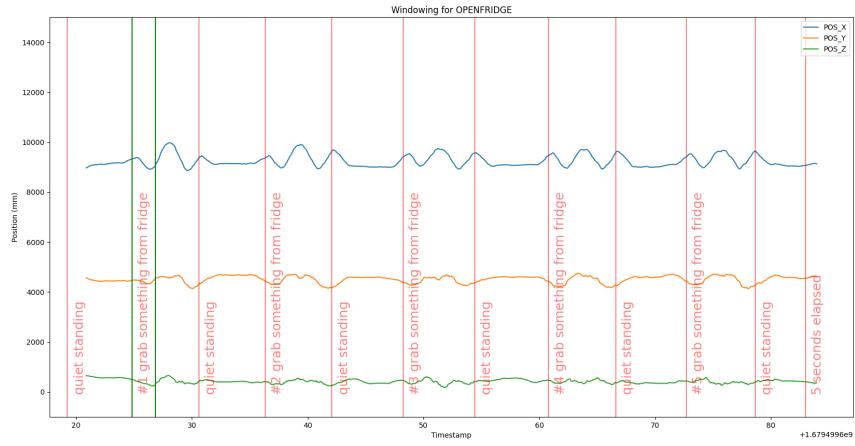
An initial sliding window with a width of 2 seconds and a stride length of 1 second was used to ensure that enough feature vectors could be extracted from the SLICETOMATO dataset. An example of windows taken for the OPENFRIDGE action and UNDEFINED action are shown in Figure 5.4.

From each window, basic statistical measures over the entire window were taken. These measures include the MEAN, MEDIAN, MODE (to 5cm for position), MAX, MIN, and STD of the entire window. From each window of data, there were a total of 3 (axes) * 5 (types of data) * 6 (statistical measures) = 90 Features

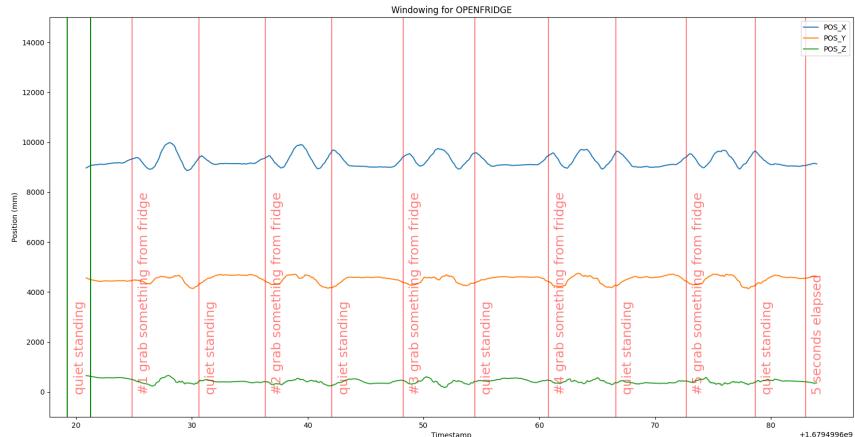
From the entire timeseries dataset, 2773 feature vectors were extracted. Refer to Table 5.1 for the breakdown of counts for each label.

Table 5.1: Count of the occurrences of each action.

Action	Count
UNDEFINED	1518
SLICETOMATO	197
WAHSHANDS	316
OPENFRIDGE	239
OPENFREEZER	203
GETPLATE	300



(a)



(b)

Figure 5.4: Obtaining windows from the OPENFRIDGE dataset. The green vertical lines section off a 2-second window. (a) A window labelled OPENFRIDGE. (b) A window labelled UNDEFINED.

5.3 Model Selection

A 60:40 split was used to train and test the model selected. Several models were chosen including Linear Support Vector Machine, Radial Support Vector Machine, K-Nearest Neighbors, Decision Trees and Random Forests. As this was a pilot study in determining the feasibility of classification of the fine-grained actions involved in making a sandwich, rigorous parameter tuning and feature selection were neglected and the defaults from the sklearn Python package were used.

5.4 Results

The confusion matrices from each model are output in Figures 5.5-5.10. Total accuracy was reported as well as the sensitivity, specificity, and precision of each class were reported. These measures are calculated as follows:

$$\text{Accuracy} = \frac{\text{All } TP}{N} \quad (5.1)$$

$$\text{Sensitivity} = \frac{TP}{TP + FN} \quad (5.2)$$

$$\text{Precision} = \frac{TP}{TP + FP} \quad (5.3)$$

$$\text{Specificity} = \frac{TN}{TN + FP} \quad (5.4)$$

Where N is the number of samples TP are True Positives, TN are true negatives, FP are false positives and FN are false negatives.

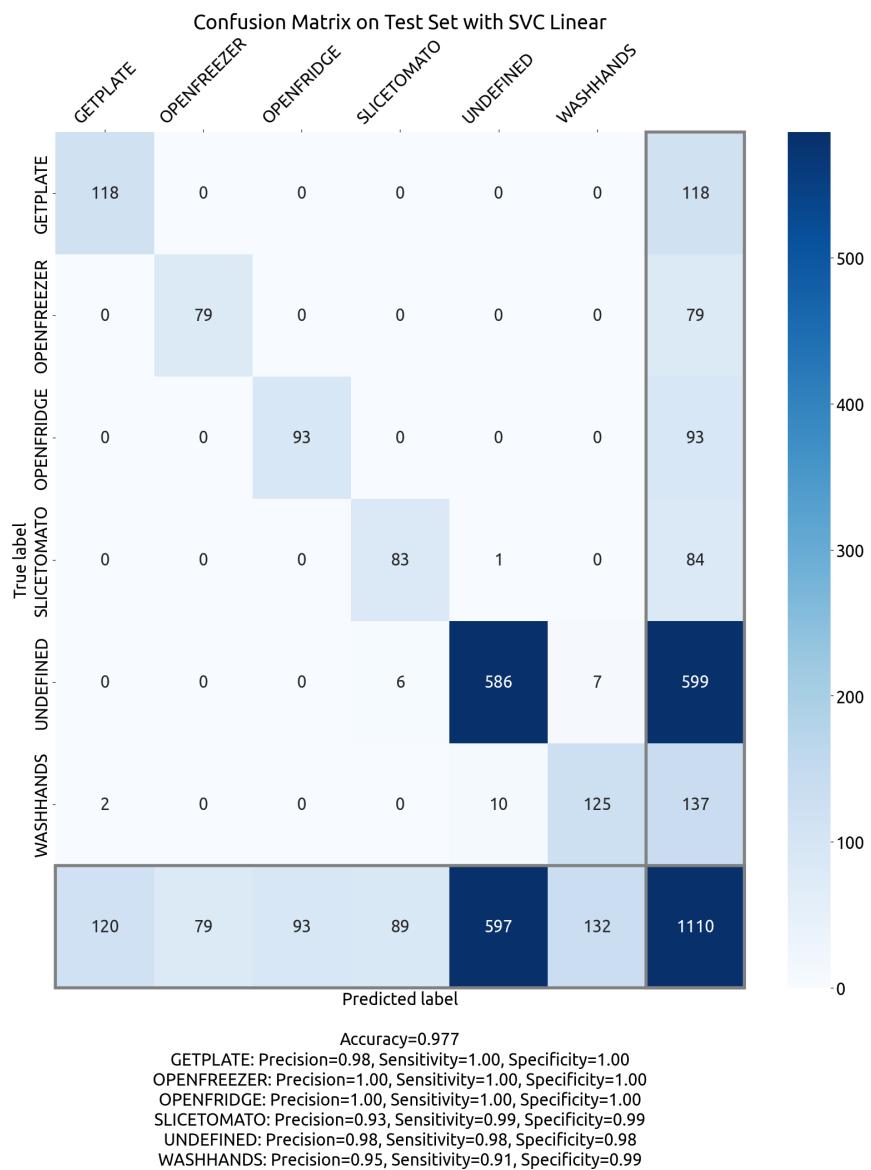


Figure 5.5: Test confusion matrix using the Support Vector Classifier with a Linear Kernel

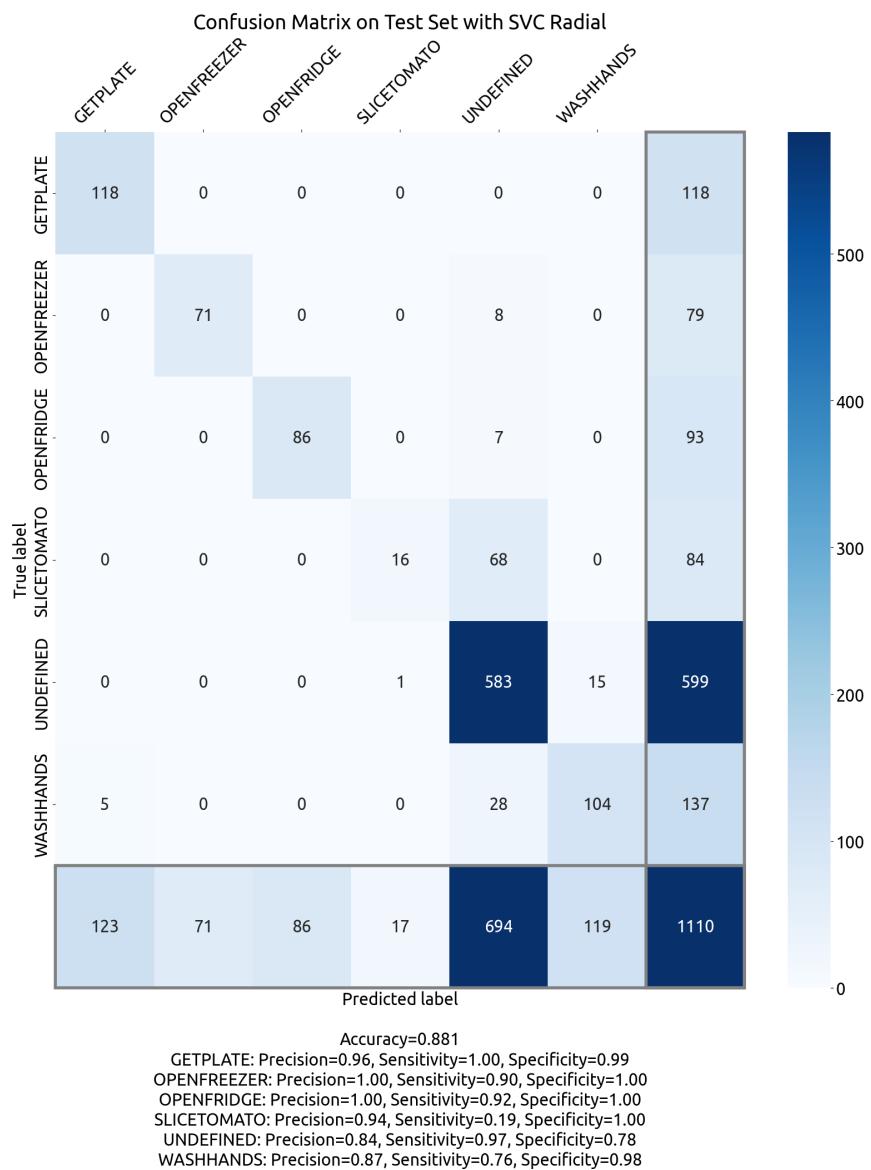


Figure 5.6: Test confusion matrix using the Support Vector Classifier with a Radial Kernel

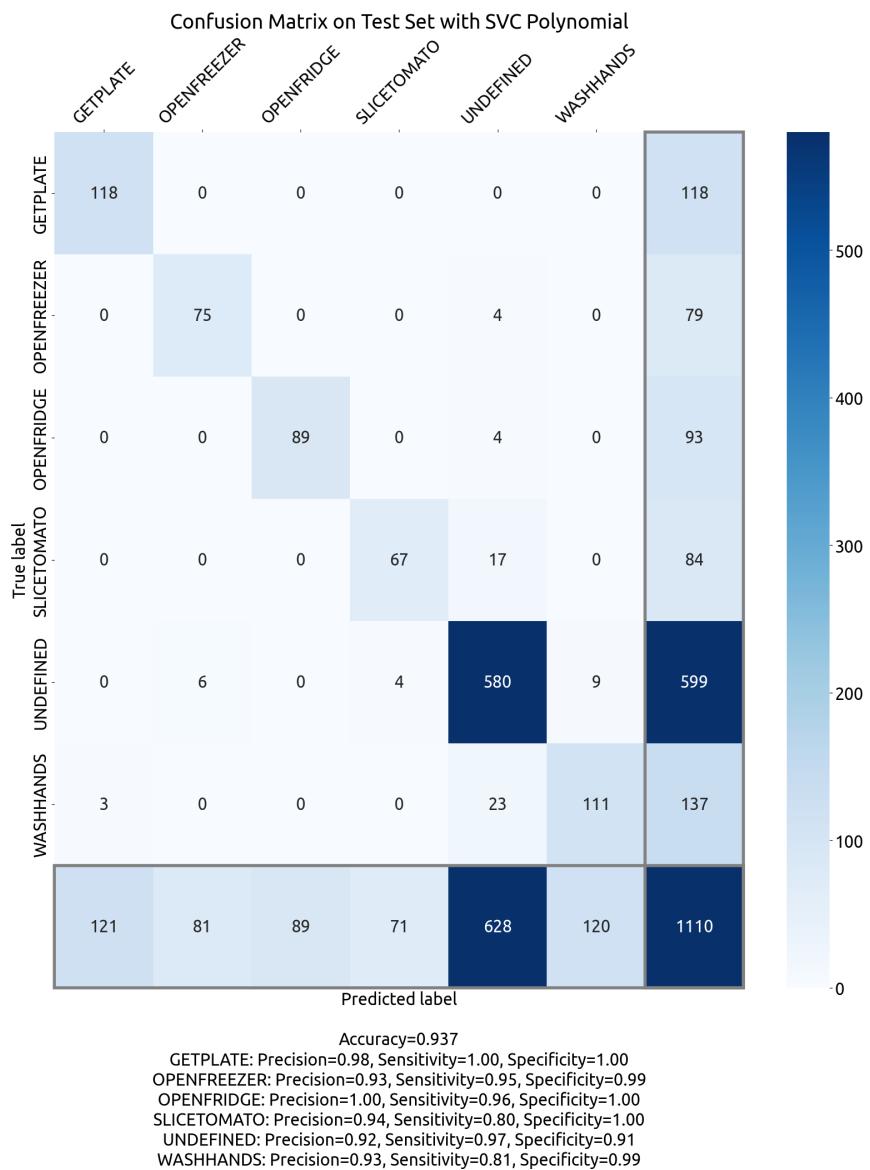


Figure 5.7: Test confusion matrix using the Support Vector Classifier with a Polynomial Kernel

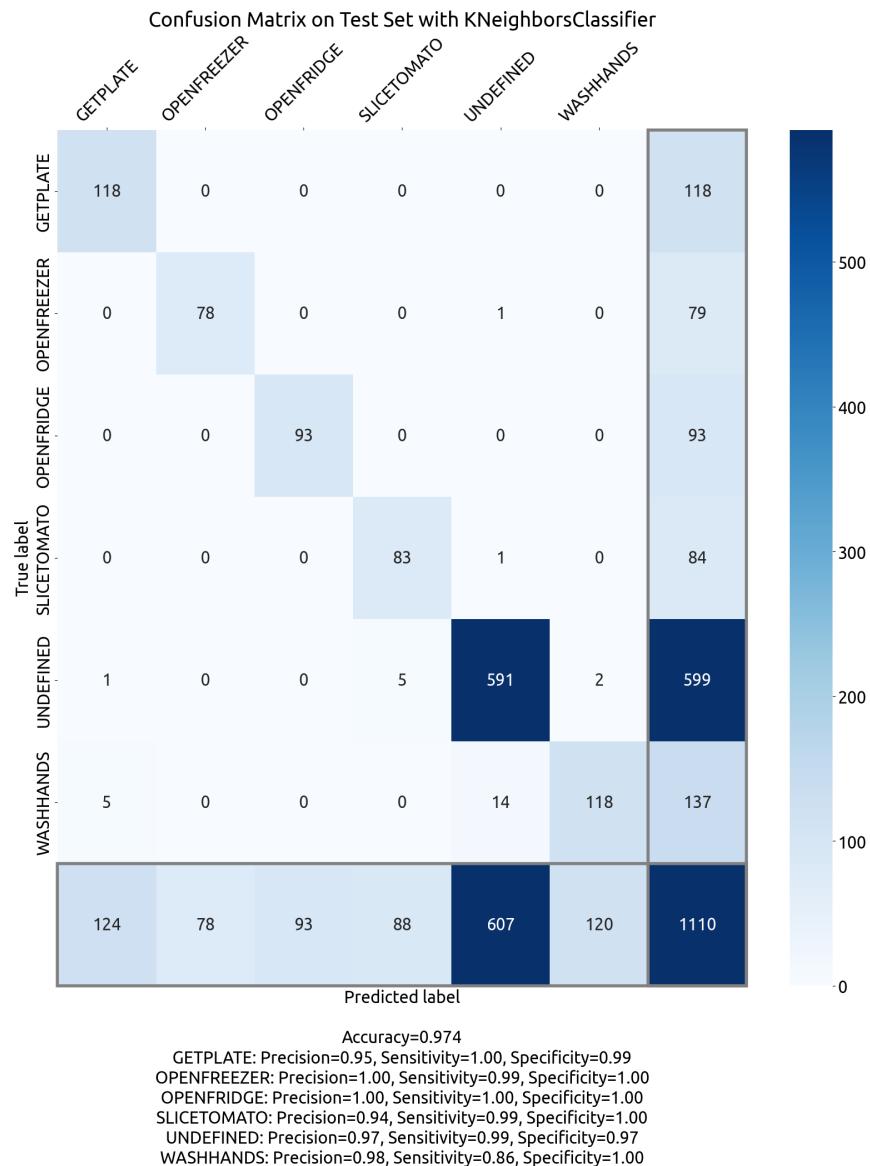


Figure 5.8: Test confusion matrix using the K-Nearest Neighbors Classifier

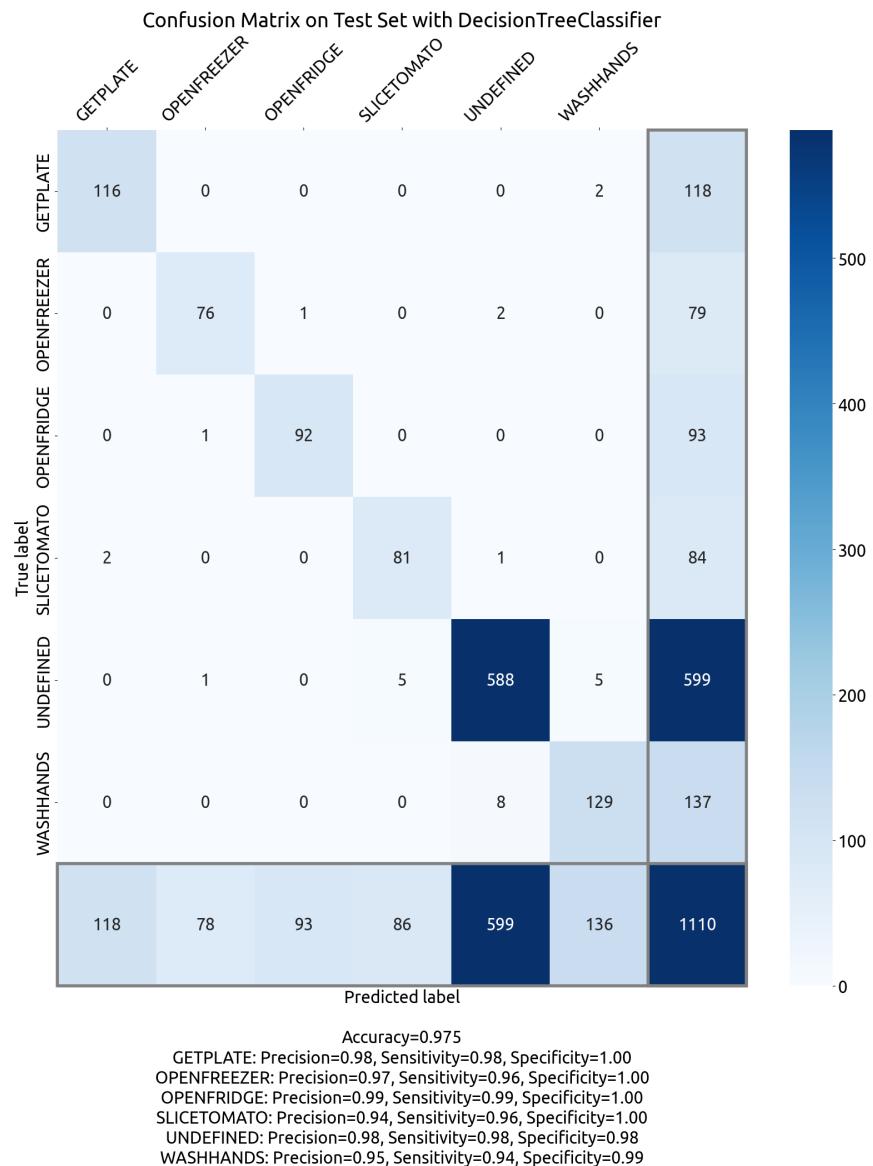


Figure 5.9: Test confusion matrix using the Decision Tree Classifier

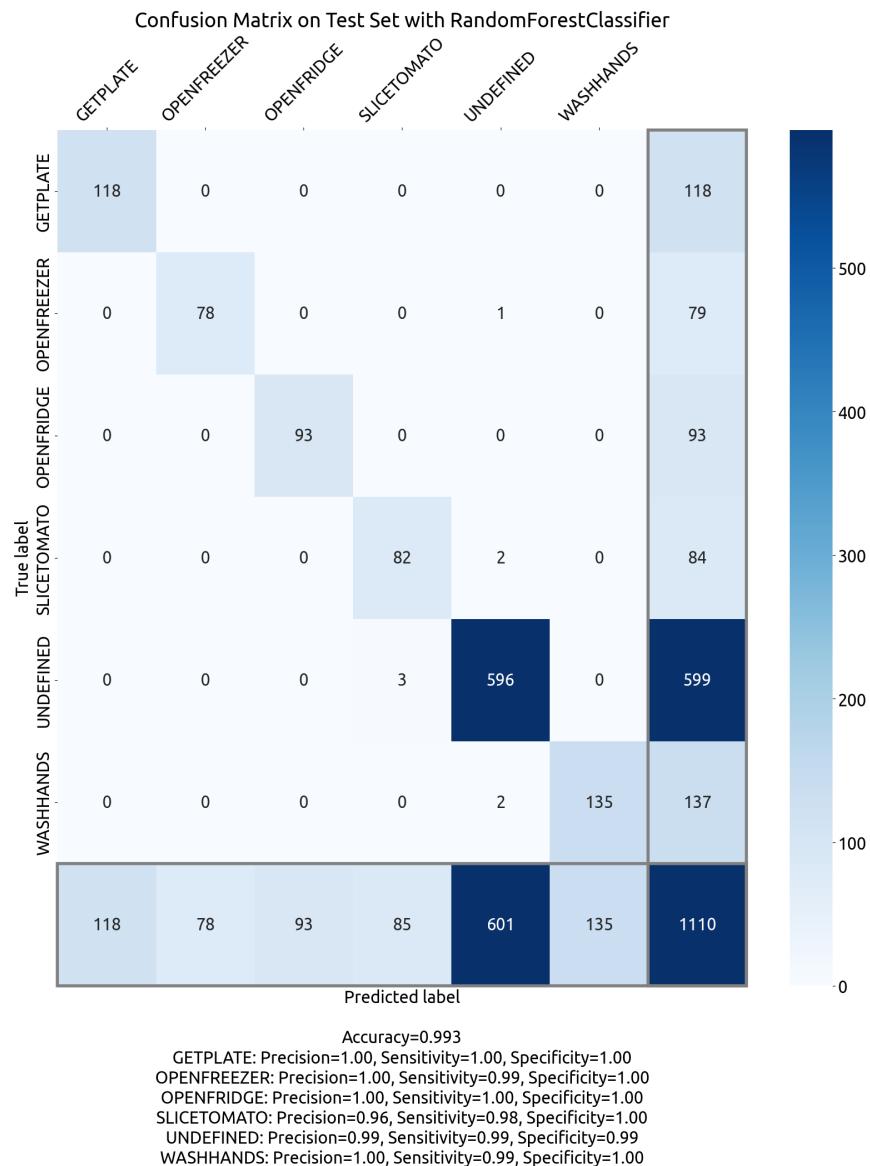


Figure 5.10: Test confusion matrix using the Random Forests Classifier

5.5 Discussion

Table 5.2 summarizes the accuracies obtained from each model.

Table 5.2: Accuracy of each Model

Model Name	Accuracy (%)
SVM Linear	97.7
SVM Radial	88.1
SVM Polynomial	93.1
kNN	97.4
Decision Tree	97.5
Random Forests	99.3

With the exception of the Radial SVM, all of the models perform well achieving an accuracy of somewhere in the high 90s. In day-to-day activities, there is a disproportionately higher number of the UNDEFINED class compared to the other "action" classes signifying the presence of class imbalance. If a classifier guesses all UNDEFINED it can obtain an accuracy of $599/1110 = 54\%$. Thus, accuracies taken around 54% should be interpreted with caution. Other metrics such as the Sensitivity, Precision and Specificity have been provided to address this class imbalance. Sensitivity is the rate at which the classifier predicts a *TP*, Precision is the fraction of predictions that are actually true, and Specificity is the rate at which the classifier predicts a *TN*. Of all the models, the Random Forests Classifier at the default settings seem to the best in terms of Accuracy and Precision, Sensitivity, and Specificity for all classes.

The performance of these models in the real-time will need to be tested and quantified before any conclusions can be made. A high accuracy is promising, but may also be indicative of overfitting which means that the model will not be able to generalize variation experienced in the real world. In later sections, more fine-grained actions will be considered, models will be more rigorously tuned, and the performance in real-time will be investigated.

Chapter 6

PASS Tasks

Bibliography

- [1] Pozyx, “Creator One Kit for research and prototyping - Pozyx.” [Online]. Available: <https://www.pozyx.io/creator-one-kit>
- [2] S. C. Government of Canada, “The Daily — Population projections: Canada, the provinces and territories, 2013 to 2063,” Sep. 2014, last Modified: 2014-09-17. [Online]. Available: <https://www150.statcan.gc.ca/n1/daily-quotidien/140917/dq140917a-eng.htm>
- [3] L. M. Tijsen, E. W. Derkx, W. P. Achterberg, and B. I. Buijck, “Challenging rehabilitation environment for older patients,” *Clinical Interventions in Aging*, vol. Volume 14, pp. 1451–1460, Aug. 2019. [Online]. Available: <https://www.dovepress.com/challenging-rehabilitation-environment-for-older-patients-peer-reviewed-article-CIA>
- [4] I. Wilkinson and A. Harper, “Comprehensive geriatric assessment, rehabilitation and discharge planning,” *Medicine*, vol. 49, no. 1, pp. 10–16, Jan. 2021. [Online]. Available: <https://linkinghub.elsevier.com/retrieve/pii/S1357303920302711>
- [5] P. F. Edemekong, D. L. Bomgaars, S. Sukumaran, and S. B. Levy, “Activities of Daily Living,” in *StatPearls*. Treasure Island (FL): StatPearls Publishing, 2022. [Online]. Available: <http://www.ncbi.nlm.nih.gov/books/NBK470404/>
- [6] E. Jaul and J. Barron, “Age-Related Diseases and Clinical and Public Health Implications for the 85 Years Old and Over Population,” *Frontiers in Public Health*, vol. 5, p. 335, Dec. 2017. [Online]. Available: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5732407/>

- [7] J. Hewitt, S. Long, B. Carter, S. Bach, K. McCarthy, and A. Clegg, “The prevalence of frailty and its association with clinical outcomes in general surgery: a systematic review and meta-analysis,” *Age and Ageing*, vol. 47, no. 6, pp. 793–800, Nov. 2018. [Online]. Available: <https://doi.org/10.1093/ageing/afy110>
- [8] S. Conroy, “Defining frailty — the holy grail of geriatric medicine,” *The Journal of Nutrition, Health and Aging*, vol. 13, no. 4, pp. 389–389, Apr. 2009. [Online]. Available: <http://link.springer.com/10.1007/s12603-009-0050-9>
- [9] X. Chen, G. Mao, and S. X. Leng, “Frailty syndrome: an overview,” *Clinical Interventions in Aging*, vol. 9, pp. 433–441, Mar. 2014. [Online]. Available: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3964027/>
- [10] “Diagnosis and Management of Dementia: Review | Dementia and Cognitive Impairment | JAMA | JAMA Network.” [Online]. Available: <https://jamanetwork.com/journals/jama/fullarticle/2753376>
- [11] I. Arevalo-Rodriguez, N. Smailagic, M. Roqué I Figuls, A. Ciapponi, E. Sanchez-Perez, A. Giannakou, O. L. Pedraza, X. Bonfill Cosp, and S. Cullum, “Mini-Mental State Examination (MMSE) for the detection of Alzheimer’s disease and other dementias in people with mild cognitive impairment (MCI),” *The Cochrane Database of Systematic Reviews*, vol. 2015, no. 3, p. CD010783, Mar. 2015.
- [12] M. Desai, L. A. Pratt, H. Lentzner, and K. N. Robinson, “Trends in vision and hearing among older Americans,” *Aging Trends (Hyattsville, Md.)*, no. 2, pp. 1–8, Mar. 2001.
- [13] J. R. Evans, A. E. Fletcher, R. P. L. Wormald, E. S.-W. Ng, S. Stirling, L. Smeeth, E. Breeze, C. J. Bulpitt, M. Nunes, D. Jones, and A. Tulloch, “Prevalence of visual impairment in people aged 75 years and older in Britain: results from the MRC trial of assessment and management of older people in the community,” *The British Journal of Ophthalmology*, vol. 86, no. 7, pp. 795–800, Jul. 2002.
- [14] B. K. Swenor, M. J. Lee, V. Varadaraj, H. E. Whitson, and P. Y. Ramulu, “Aging With Vision Loss: A Framework for Assessing the

- Impact of Visual Impairment on Older Adults,” *The Gerontologist*, vol. 60, no. 6, pp. 989–995, Aug. 2020. [Online]. Available: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7427480/>
- [15] M. E. Tinetti, C. S. Williams, and T. M. Gill, “Dizziness among older adults: a possible geriatric syndrome,” *Annals of Internal Medicine*, vol. 132, no. 5, pp. 337–344, Mar. 2000.
 - [16] S. Iwasaki and T. Yamasoba, “Dizziness and Imbalance in the Elderly: Age-related Decline in the Vestibular System,” *Aging and Disease*, vol. 6, no. 1, pp. 38–47, Feb. 2014. [Online]. Available: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4306472/>
 - [17] G. Savarese and L. H. Lund, “Global Public Health Burden of Heart Failure,” *Cardiac Failure Review*, vol. 3, no. 1, pp. 7–11, Apr. 2017. [Online]. Available: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5494150/>
 - [18] B. Ziaeian and G. C. Fonarow, “Epidemiology and aetiology of heart failure,” *Nature Reviews. Cardiology*, vol. 13, no. 6, pp. 368–378, Jun. 2016.
 - [19] R. D. S. Watson, C. R. Gibbs, and G. Y. H. Lip, “Clinical features and complications,” *BMJ : British Medical Journal*, vol. 320, no. 7229, pp. 236–239, Jan. 2000. [Online]. Available: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC1117436/>
 - [20] I. o. M. U. C. o. S. S. C. D. Criteria, “Ischemic Heart Disease,” in *Cardiovascular Disability: Updating the Social Security Listings*. National Academies Press (US), 2010. [Online]. Available: <https://www.ncbi.nlm.nih.gov/books/NBK209964/>
 - [21] M. V. Madhavan, B. J. Gersh, K. P. Alexander, C. B. Granger, and G. W. Stone, “Coronary Artery Disease in Patients Å ª 80 Years of Age,” *Journal of the American College of Cardiology*, vol. 71, no. 18, pp. 2015–2040, May 2018. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0735109718336167>
 - [22] A. Sapra and P. Bhandari, “Diabetes Mellitus,” in *StatPearls*. Treasure Island (FL): StatPearls Publishing, 2021. [Online]. Available: <http://www.ncbi.nlm.nih.gov/books/NBK551501/>

- [23] “Glucose tolerance tests: What exactly do they involve?” in *InformedHealth.org [Internet]*. Institute for Quality and Efficiency in Health Care (IQWiG), Oct. 2020. [Online]. Available: <https://www.ncbi.nlm.nih.gov/books/NBK279331/>
- [24] M. J. Lespasio, A. A. Sultan, N. S. Piuzzi, A. Khlopas, M. E. Husni, G. F. Muschler, and M. A. Mont, “Hip Osteoarthritis: A Primer,” *The Permanente Journal*, vol. 22, pp. 17–084, Jan. 2018. [Online]. Available: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5760056/>
- [25] J. M. Hootman and C. G. Helmick, “Projections of US prevalence of arthritis and associated activity limitations,” *Arthritis and Rheumatism*, vol. 54, no. 1, pp. 226–229, Jan. 2006.
- [26] D. Prieto-Lambra, D. Hunter, and N. Arden, *Osteoarthritis: The Facts*, second edition ed., ser. The Facts Series. Oxford, New York: Oxford University Press, Sep. 2014.
- [27] D. J. Hunter and S. Bierma-Zeinstra, “Osteoarthritis,” *The Lancet*, vol. 393, no. 10182, pp. 1745–1759, Apr. 2019. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0140673619304179>
- [28] R. Altman, E. Asch, D. Bloch, G. Bole, D. Borenstein, K. Brandt, W. Christy, T. D. Cooke, R. Greenwald, and M. Hochberg, “Development of criteria for the classification and reporting of osteoarthritis. Classification of osteoarthritis of the knee. Diagnostic and Therapeutic Criteria Committee of the American Rheumatism Association,” *Arthritis and Rheumatism*, vol. 29, no. 8, pp. 1039–1049, Aug. 1986.
- [29] J. E. Compston, M. R. McClung, and W. D. Leslie, “Osteoporosis,” *The Lancet*, vol. 393, no. 10169, pp. 364–376, Jan. 2019, publisher: Elsevier. [Online]. Available: [https://www-thelancet-com.login.ezproxy.library.ualberta.ca/journals/lancet/article/PIIS0140-6736\(18\)32112-3/fulltext](https://www-thelancet-com.login.ezproxy.library.ualberta.ca/journals/lancet/article/PIIS0140-6736(18)32112-3/fulltext)
- [30] J. L. Porter and M. Varacallo, “Osteoporosis,” in *StatPearls*. Treasure Island (FL): StatPearls Publishing, 2023. [Online]. Available: <http://www.ncbi.nlm.nih.gov/books/NBK441901/>
- [31] D. L. Glaser and F. S. Kaplan, “Osteoporosis: Definition and Clinical Presentation,” *Spine*, vol. 22, no. 24, p. 12S, Dec. 1997.

- [32] M. Pashmdarfard and A. Azad, “Assessment tools to evaluate Activities of Daily Living (ADL) and Instrumental Activities of Daily Living (IADL) in older adults: A systematic review,” *Medical Journal of the Islamic Republic of Iran*, vol. 34, p. 33, Apr. 2020. [Online]. Available: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7320974/>
- [33] E. McMahon, “Katz Index of Independence in Activities of Daily Living,” p. 2.
- [34] P. P. Katz for the Association of Rheumatology Health Professionals Outcomes Measures Task Force, “Measures of adult general functional status: The Barthel Index, Katz Index of Activities of Daily Living, Health Assessment Questionnaire (HAQ), MACTAR Patient Preference Disability Questionnaire, and Modified Health Assessment Questionnaire (MHAQ),” *Arthritis Care & Research*, vol. 49, no. S5, pp. S15–S27, 2003, eprint: <https://onlinelibrary.wiley.com/doi/pdf/10.1002/art.11415>. [Online]. Available: <https://onlinelibrary.wiley.com/doi/abs/10.1002/art.11415>
- [35] AHS, “Barthel Index of Activities of Daily Living.”
- [36] S. Katz, A. B. Ford, R. W. Moskowitz, B. A. Jackson, and M. W. Jaffe, “Studies of Illness in the Aged: The Index of ADL: A Standardized Measure of Biological and Psychosocial Function,” *JAMA*, vol. 185, no. 12, pp. 914–919, Sep. 1963. [Online]. Available: <https://doi.org/10.1001/jama.1963.03060120024016>
- [37] M. Davidson, “Functional Independence Measure,” in *Encyclopedia of Quality of Life and Well-Being Research*, A. C. Michalos, Ed. Dordrecht: Springer Netherlands, 2014, pp. 2373–2376.
- [38] “FIM(TM).” [Online]. Available: <https://www.tbims.org/combi/FIM/>
- [39] E. Dutil, A. Forget, M. Vanier, and C. Gaudreault, “Development of the ADL Profile: An Evaluation for Adults with Severe Head Injury,” *Occupational Therapy In Health Care*, vol. 7, no. 1, pp. 7–22, Jan. 1990.
- [40] C. L. Bottari, C. Dassa, C. M. Rainville, and E. Dutil, “The IADL profile: development, content validity, intra- and interrater agreement,” *Canadian Journal of Occupational Therapy. Revue Canadienne D'ergotherapie*, vol. 77, no. 2, pp. 90–100, Apr. 2010.

- [41] N. Kennedy, A. Barion, A. Rademaker, G. Rehkemper, and S. Weintraub, “The Activities of Daily Living Questionnaire A Validation Study in Patients with Dementia,” *Alzheimer disease and associated disorders*, vol. 18, pp. 223–30, Oct. 2004.
- [42] A. Perry, M. Morris, C. Unsworth, S. Duckett, J. Skeat, K. Dodd, N. Taylor, and K. Reilly, “Therapy outcome measures for allied health practitioners in Australia: the AusTOMs,” *International Journal for Quality in Health Care*, vol. 16, no. 4, pp. 285–291, Aug. 2004. [Online]. Available: <https://doi.org/10.1093/intqhc/mzh059>
- [43] S. A. Haymes, A. W. Johnston, and A. D. Heyes, “The Development of the Melbourne Low-Vision ADL Index: A Measure of Vision Disability,” *Investigative Ophthalmology & Visual Science*, vol. 42, no. 6, pp. 1215–1225, May 2001.
- [44] R. G. Brown, B. MacCarthy, M. Jahanshahi, and C. D. Marsden, “Accuracy of Self-Reported Disability in Patients With Parkinsonism,” *Archives of Neurology*, vol. 46, no. 9, pp. 955–959, Sep. 1989. [Online]. Available: <https://doi.org/10.1001/archneur.1989.00520450025014>
- [45] M. Holbrook and C. E. Skilbeck, “AN ACTIVITIES INDEX FOR USE WITH STROKE PATIENTS,” *Age and Ageing*, vol. 12, no. 2, pp. 166–170, 1983. [Online]. Available: <https://academic.oup.com/ageing/article-lookup/doi/10.1093/ageing/12.2.166>
- [46] M. P. Lawton and E. M. Brody, “Assessment of older people: self-maintaining and instrumental activities of daily living,” *The Gerontologist*, vol. 9, no. 3, pp. 179–186, 1969.
- [47] J. C. Rogers, “Performance Assessment of Self-Care Skills,” Apr. 2014. [Online]. Available: <http://doi.apa.org/getdoi.cfm?doi=10.1037/t16068-000>
- [48] D. Chisholm, P. Toto, K. Raina, M. Holm, and J. Rogers, “Evaluating capacity to live independently and safely in the community: Performance Assessment of Self-care Skills,” *The British journal of occupational therapy*, vol. 77, no. 2, pp. 59–63, Feb. 2014. [Online]. Available: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4186770/>

- [49] C. M. Cullum, K. Saine, L. D. Chan, K. Martin-Cook, K. F. Gray, and M. F. Weiner, “Performance-Based instrument to assess functional capacity in dementia: The Texas Functional Living Scale,” *Neuropsychiatry, Neuropsychology, and Behavioral Neurology*, vol. 14, no. 2, pp. 103–108, 2001.
- [50] A. Staff, “The (Original) Barthel Index of ADLs,” Sep. 2008. [Online]. Available: <https://www.elitelearning.com/resource-center/rehabilitation-therapy/the-original-barthel-index-of-adls/>
- [51] “Measures of adult general functional status: The Barthel Index, Katz Index of Activities of Daily Living, Health Assessment Questionnaire (HAQ), MACTAR Patient Preference Disability Questionnaire, and Modified Health Assessment Questionnaire (MHAQ).” [Online]. Available: <https://onlinelibrary.wiley.com/doi/10.1002/art.11415>
- [52] K. M. Hall, B. B. Hamilton, W. A. Gordon, and N. D. Zasler, “Functional Independence Measure and Functional Assessment Measure,” Jun. 2014, institution: American Psychological Association. [Online]. Available: <http://doi.apa.org/getdoi.cfm?doi=10.1037/t28782-000>
- [53] “ADL Profile – Strokengine.” [Online]. Available: <https://strokengine.ca/en/assessments/adl/>
- [54] cunsworth2014, “AusTOMs,” Nov. 2019. [Online]. Available: <https://austoms.com/>
- [55] “Self-assessment Parkinsonâ€™s Disease Disability Scale | RehabMeasures Database,” May 2013. [Online]. Available: <https://www.sralab.org/rehabilitation-measures/self-assessment-parkinsons-disease-disability-scale>
- [56] J. Schuling, R. De Haan, M. Limburg, and K. H. Groenier, “The Frenchay Activities Index. Assessment of functional status in stroke patients.” *Stroke*, vol. 24, no. 8, pp. 1173–1177, Aug. 1993. [Online]. Available: <https://www.ahajournals.org/doi/10.1161/01.STR.24.8.1173>
- [57] E. McMahon, “Lawton â€“ Brody Instrumental Activities of Daily Living Scale (IADL).”

- [58] “Performance Assessment of Self-Care Skills | RehabMeasures Database,” Jun. 2015. [Online]. Available: <https://www.sralab.org/rehabilitation-measures/performance-assessment-self-care-skills>
- [59] U. F. O. Themes, “Texas Functional Living Scale (TFLS),” Jul. 2017. [Online]. Available: <https://nursekey.com/texas-functional-living-scale-tfls/>
- [60] N. Camp, M. Lewis, K. Hunter, J. Johnston, M. Zecca, A. Di Nuovo, and D. Magistro, “Technology Used to Recognize Activities of Daily Living in Community-Dwelling Older Adults,” *International Journal of Environmental Research and Public Health*, vol. 18, no. 1, p. 163, Jan. 2021, number: 1 Publisher: Multidisciplinary Digital Publishing Institute. [Online]. Available: <https://www.mdpi.com/1660-4601/18/1/163>
- [61] N. Gadey, P. Pataunia, A. Chan, and A. RÃos RincÃ³n, “Technologies for monitoring activities of daily living in older adults: a systematic review,” *Disability and Rehabilitation. Assistive Technology*, pp. 1–10, Mar. 2023.
- [62] S. A. Sikkes and J. d. Rotrou, “A qualitative review of instrumental activities of daily living in dementia: what’s cooking?” *Neurodegenerative Disease Management*, vol. 4, no. 5, pp. 393–400, Oct. 2014. [Online]. Available: <https://www.futuremedicine.com/doi/10.2217/nmt.14.24>
- [63] B. Bouchard, K. Bouchard, and A. Bouzouane, “A smart cooking device for assisting cognitively impaired users,” *Journal of Reliable Intelligent Environments*, vol. 6, no. 2, pp. 107–125, Jun. 2020. [Online]. Available: <https://link.springer.com/10.1007/s40860-020-00104-3>
- [64] E. Dubuc, M. Gagnon-Roy, M. Couture, N. Bier, S. Giroux, and C. Bottari, “Perceived needs and difficulties in meal preparation of people living with traumatic brain injury in a chronic phase: Supporting long-term services and interventions,” *Australian Occupational Therapy Journal*, vol. 66, no. 6, pp. 720–730, 2019, eprint: <https://onlinelibrary.wiley.com/doi/pdf/10.1111/1440-1630.12611> [Online]. Available: <https://onlinelibrary.wiley.com/doi/abs/10.1111/1440-1630.12611>

- [65] P. Malani, J. Kullgren, E. Solway, J. Wolfson, C. Leung, D. Singer, and M. Kirch, “The Joy of Cooking and its Benefits for Older Adults,” University of Michigan, Tech. Rep., Jun. 2020.
- [66] R. Yared, B. Abdulrazak, T. Tessier, and P. Mabilleau, “Cooking risk analysis to enhance safety of elderly people in smart kitchen,” in *Proceedings of the 8th ACM International Conference on PErvasive Technologies Related to Assistive Environments*. Corfu Greece: ACM, Jul. 2015, pp. 1–4. [Online]. Available: <https://dl.acm.org/doi/10.1145/2769493.2769516>
- [67] D. J. Cook, “Learning Setting-Generalized Activity Models for Smart Spaces,” *IEEE intelligent systems*, vol. 2010, no. 99, p. 1, Sep. 2010. [Online]. Available: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3068197/>
- [68] B. Logan and J. Healey, “Sensors to Detect the Activities of Daily Living,” in *2006 International Conference of the IEEE Engineering in Medicine and Biology Society*, Aug. 2006, pp. 5362–5365, iSSN: 1557-170X.
- [69] N. Sarma, S. Chakraborty, and D. S. Banerjee, “Activity Recognition through Feature Learning and Annotations using LSTM,” in *2019 11th International Conference on Communication Systems & Networks (COMSNETS)*. Bengaluru, India: IEEE, Jan. 2019, pp. 444–447. [Online]. Available: <https://ieeexplore.ieee.org/document/8711147/>
- [70] M. Mokhtari, B. Abdulrazak, and H. Aloulou, Eds., *Smart Homes and Health Telematics, Designing a Better Future: Urban Assisted Living: 16th International Conference, ICOST 2018, Singapore, Singapore, July 10-12, 2018, Proceedings*, ser. Lecture Notes in Computer Science. Cham: Springer International Publishing, 2018, vol. 10898. [Online]. Available: <http://link.springer.com/10.1007/978-3-319-94523-1>
- [71] K. Yordanova, S. Lüdtke, S. Whitehouse, F. Krüger, A. Paiement, M. Mirmehdi, I. Craddock, and T. Kirste, “Analysing Cooking Behaviour in Home Settings: Towards Health Monitoring,” *Sensors*, vol. 19, no. 3, p. 646, Jan. 2019, number: 3 Publisher: Multidisciplinary Digital Publishing Institute. [Online]. Available: <https://www.mdpi.com/1424-8220/19/3/646>

- [72] D. J. Cook and M. Schmitter-Edgecombe, “Assessing the Quality of Activities in a Smart Environment,” *Methods of Information in Medicine*, vol. 48, no. 05, pp. 480–485, 2009. [Online]. Available: <http://www.thieme-connect.de/DOI/DOI?10.3414/ME0592>
- [73] P. N. Dawadi, D. J. Cook, M. Schmitter-Edgecombe, and C. Parsey, “Automated Assessment of Cognitive Health Using Smart Home Technologies,” *Technology and health care : official journal of the European Society for Engineering and Medicine*, vol. 21, no. 4, pp. 323–343, 2013. [Online]. Available: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4143248/>
- [74] P.-W. Chen, N. A. Baune, I. Zwir, J. Wang, V. Swamidass, and A. W. Wong, “Measuring Activities of Daily Living in Stroke Patients with Motion Machine Learning Algorithms: A Pilot Study,” *International Journal of Environmental Research and Public Health*, vol. 18, no. 4, p. 1634, Feb. 2021. [Online]. Available: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7915561/>
- [75] S. Pan, M. Berges, J. Rodakowski, P. Zhang, and H. Y. Noh, “Fine-Grained Activity of Daily Living (ADL) Recognition Through Heterogeneous Sensing Systems With Complementary Spatiotemporal Characteristics,” *Frontiers in Built Environment*, vol. 6, 2020. [Online]. Available: <https://www.frontiersin.org/articles/10.3389/fbuil.2020.560497>
- [76] J. Chung, G. Demiris, and H. J. Thompson, “Ethical Considerations Regarding the Use of Smart Home Technologies for Older Adults: An Integrative Review,” *Annual Review of Nursing Research*, vol. 34, no. 1, pp. 155–181, Jan. 2016. [Online]. Available: <http://connect.springerpub.com/lookup/doi/10.1891/0739-6686.34.155>
- [77] G. Demiris, “Privacy and social implications of distinct sensing approaches to implementing smart homes for older adults,” *Annual International Conference of the IEEE Engineering in Medicine and Biology Society. IEEE Engineering in Medicine and Biology Society. Annual International Conference*, vol. 2009, pp. 4311–4314, 2009.

- [78] J. Leo and D. Goodwin, “Negotiated Meanings of Disability Simulations in an Adapted Physical Activity Course: Learning From Student Reflections,” *Adapted Physical Activity Quarterly*, vol. 31, no. 2, pp. 144–161, Apr. 2014. [Online]. Available: <https://journals.humankinetics.com/view/journals/apaq/31/2/article-p144.xml>
- [79] Pozyx, “Hardware setup.” [Online]. Available: <https://docs.pozyx.io/creator/hardware-setup>
- [80] ——, “Configuration of the UWB parameters (Arduino).” [Online]. Available: <https://docs.pozyx.io/creator/configuration-of-the-uwb-parameters-arduino>
- [81] Z. Zhang, “Introduction to machine learning: K-nearest neighbors,” vol. 4, no. 11, p. 218. [Online]. Available: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4916348/>
- [82] Scipy.stats.mode — SciPy v0.7 Reference Guide (DRAFT). [Online]. Available: <https://docs.scipy.org/doc/scipy-0.7.x/reference/generated/scipy.stats.mode.html>
- [83] J. Faouzi, “Time Series Classification: A Review of Algorithms and Implementations,” in *Time Series Analysis - Recent Advances, New Perspectives and Applications*, J. Rocha, C. M. Viana, and S. Oliveira, Eds. IntechOpen. [Online]. Available: <https://www.intechopen.com/chapters/1185930>
- [84] J. Cervantes, F. Garcia-Lamont, L. Rodríguez-Mazahua, and A. Lopez, “A comprehensive survey on support vector machine classification: Applications, challenges and trends,” vol. 408, pp. 189–215. [Online]. Available: <https://linkinghub.elsevier.com/retrieve/pii/S0925231220307153>
- [85] 1.12. Multiclass and multioutput algorithms. scikit-learn. [Online]. Available: <https://scikit-learn/stable/modules/multiclass.html>
- [86] Y. Liu, Y. Wang, and J. Zhang, “New Machine Learning Algorithm: Random Forest,” in *Information Computing and Applications*, B. Liu, M. Ma, and J. Chang, Eds. Springer, pp. 246–252.

- [87] V. E. Lee and L. Liu, “Decision Trees: Theory and Algorithms.”
- [88] L. Breiman, “Random Forests,” vol. 45, no. 1, pp. 5–32. [Online]. Available: <https://doi.org/10.1023/A:1010933404324>
- [89] 1.10. Decision Trees. scikit-learn. [Online]. Available: <https://scikit-learn/stable/modules/tree.html>
- [90] Scikit-learn/sklearn/tree/splitter.pyx at main · scikit-learn/scikit-learn · GitHub. [Online]. Available: https://github.com/scikit-learn/scikit-learn/blob/0b0b90b75102226d27e3947b9766729325d7042c/sklearn/tree/_splitter.pyx
- [91] RandomForestClassifier. scikit-learn. [Online]. Available: <https://scikit-learn/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html>
- [92] P. Probst, M. Wright, and A.-L. Boulesteix, “Hyperparameters and Tuning Strategies for Random Forest,” vol. 9, no. 3, p. e1301. [Online]. Available: <http://arxiv.org/abs/1804.03515>
- [93] L. Ye and E. Keogh, “Time series shapelets: A new primitive for data mining,” in *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, pp. 947–956. [Online]. Available: <https://dl.acm.org/doi/10.1145/1557019.1557122>
- [94] D. A. Roberts, S. Yaida, and B. Hanin, *The Principles of Deep Learning Theory*. [Online]. Available: <http://arxiv.org/abs/2106.10165>
- [95] What is a Neural Network? — IBM. [Online]. Available: <https://www.ibm.com/topics/neural-networks>
- [96] M. M. Taye, “Theoretical Understanding of Convolutional Neural Network: Concepts, Architectures, Applications, Future Directions,” vol. 11, no. 3, p. 52. [Online]. Available: <https://www.mdpi.com/2079-3197/11/3/52>
- [97] K. B. Prakash, R. Kannan, S. Alexander, and G. R. Kanagachidambaresan, Eds., *Advanced Deep Learning for Engineers and Scientists: A Practical Approach*, ser. EAI/Springer Innovations in Communication and Computing. Springer International Publishing. [Online]. Available: <https://link.springer.com/10.1007/978-3-030-66519-7>

- [98] P. Purwono, A. Ma’arif, W. Rahmaniар, H. Imam, H. I. K. Fathurrahman, A. Frisky, and Q. M. U. Haq, “Understanding of Convolutional Neural Network (CNN): A Review,” vol. 2, pp. 739–748.
- [99] L. Alzubaidi, J. Zhang, A. J. Humaidi, A. Al-Dujaili, Y. Duan, O. Al-Shamma, J. Santamaría, M. A. Fadhel, M. Al-Amidie, and L. Farhan, “Review of deep learning: Concepts, CNN architectures, challenges, applications, future directions,” vol. 8, no. 1, p. 53. [Online]. Available: <https://doi.org/10.1186/s40537-021-00444-8>
- [100] M. Swapna, D. Y. Sharma, and B. Prasad, “CNN Architectures: Alex Net, Le Net, VGG, Google Net, Res Net,” vol. 8, pp. 953–959.
- [101] Training a Classifier — PyTorch Tutorials 2.4.0+cu121 documentation. [Online]. Available: https://pytorch.org/tutorials/beginner/blitz/cifar10_tutorial.html
- [102] R. Vasudev. Understanding and Calculating the number of Parameters in Convolution Neural Networks (CNNs). Medium. [Online]. Available: <https://towardsdatascience.com/understanding-and-calculating-the-number-of-parameters-in-convolution-neural-networks-cnns-fc88790d530d>
- [103] N. Zakaria. Backpropagation in CNN; A Mathematically Explicit Exposition. Medium. [Online]. Available: <https://nordinzakaria.medium.com/backpropagation-in-cnn-a-mathematically-explicit-exposition-3f723f0ad9a0>
- [104] A. Sherstinsky, “Fundamentals of Recurrent Neural Network (RNN) and Long Short-Term Memory (LSTM) network,” vol. 404, p. 132306. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0167278919305974>
- [105] F. M. Salem, *Recurrent Neural Networks: From Simple to Gated Architectures*. Springer International Publishing. [Online]. Available: <https://link.springer.com/10.1007/978-3-030-89929-5>
- [106] CS231n Convolutional Neural Networks for Visual Recognition. [Online]. Available: <https://cs231n.github.io/rnn/>

- [107] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin. Attention Is All You Need. [Online]. Available: <http://arxiv.org/abs/1706.03762>
- [108] Torchtext.data.utils — Torchtext 0.18.0 documentation. [Online]. Available: https://pytorch.org/text/stable/data_utils.html
- [109] T. H. Teng, K. D. Varathan, and F. Crestani, “A comprehensive review of cyberbullying-related content classification in online social media,” vol. 244, p. 122644. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0957417423031469>
- [110] Amanatullah. The Attention Mechanism. Medium. [Online]. Available: <https://medium.com/@amanatulla1606/the-attention-mechanism-797e51c45d46>
- [111] Longx. Transformer Attention Layer gradient. Longxiang He. [Online]. Available: <https://say-hello2y.github.io/2022-09-07/attention-gradient>
- [112] Y. Yin, “Prediction and analysis of time series data based on granular computing,” vol. 17. [Online]. Available: <https://www.frontiersin.org/journals/computational-neuroscience/articles/10.3389/fncom.2023.1192876/full>
- [113] Z. Wang, W. Yan, and T. Oates. Time Series Classification from Scratch with Deep Neural Networks: A Strong Baseline. [Online]. Available: <http://arxiv.org/abs/1611.06455>
- [114] LSTM — PyTorch 2.5 documentation. [Online]. Available: <https://pytorch.org/docs/stable/generated/torch.nn.LSTM.html>
- [115] M. Barandas, D. Folgado, L. Fernandes, S. Santos, M. Abreu, P. Bota, H. Liu, T. Schultz, and H. Gamboa, “TSFEL: Time Series Feature Extraction Library,” vol. 11, p. 100456. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S2352711020300017>
- [116] List of available features — TSFEL 0.1.9 documentation. [Online]. Available: https://tsfel.readthedocs.io/en/latest/descriptions/feature_list.html