TOWARDS MULTIMODAL ACTIVITY RECOGNITION IN COMPLEX SCENARIOS

Inauguraldissertation
zur Erlangung des akademischen Grades
eines Doktors der Naturwissenschaften
der Universität Mannheim

vorgelegt von

Alexander Diete
aus Tschuchlomino, Russland

Mannheim, 2021

Dekan: Dr. Bernd Lübcke, Universität Mannheim
Referent: Professor Dr. Heiner Stuckenschmidt, Universität Mannheim
Korreferent: Professor Dr. Margret Keuper, Universität Mannheim

Tag der mündlichen Prüfung: 25.02.2021

Eidesstattliche Versicherung gemäß § 7 Absatz 2 Buchstabe c) der Promotionsordnung der Universität Mannheim zur Erlangung des Doktorgrades der Naturwissenschaften:

i Bei der eingereichten Dissertation zum Thema "Towards Multimodal Activity Recognition In Complex Scenarios" handelt es sich um mein eigenständig erstelltes eigenes Werk.

ii Ich habe nur die angegebenen Quellen und Hilfsmittel benutzt und mich keiner unzulässigen Hilfe Dritter bedient. Insbesondere habe ich wörtliche Zitate aus anderen Werken als solche kenntlich gemacht.

iii Die Arbeit oder Teile davon habe ich bislang nicht an einer Hochschule des In- oder Auslands als Bestandteil einer Prüfungs- oder Qualifikationsleistung vorgelegt. Die Richtigkeit der vorstehenden Erklärung bestätige ich.

iv Die Bedeutung der eidesstattlichen Versicherung und die strafrechtlichen Folgen einer unrichtigen oder unvollständigen eidesstattlichen Versicherung sind mir bekannt.

Ich versichere an Eides statt, dass ich nach bestem Wissen die reine Wahrheit erklärt und nichts verschwiegen habe.

ABSTRACT

Activity recognition deals with the task of figuring out a person's current activity based on the reading of sensors. Many downstream applications can utilize this information to aid potential users. Fitness tracking devices, for instance, try to determine the timespan a user was running or riding a bike to aid workout goals. Other applications can be found in the health sector, where activity recognition can be used to gain insights about patients that have cognitive disabilities and may forget important daily activities like food or medicine consumption. It can also aid logistics by streamlining processes, e.g. in order picking scenarios for warehouses. For many of these applications, either one or multiple sensors are utilized. In this work, we look at multimodal activity recognition, meaning we consider multiple sensors of different types. We believe that the combination of different modalities can improve the overall accuracy of current methods. Since sensors are becoming cheaper and smaller, practical implementation of multi-sensor systems become more feasible. Specifically, we use video and inertial information for our methods. With this setting in mind, our contribution is spread throughout the whole activity recognition pipeline. We create datasets, develop new annotation methods, and evaluate new types of features and models for activity recognition in both industrial and personal use.

ZUSAMMENFASSUNG

Aktivitätserkennung ist ein Feld in der Informatik, dass sich mit dem Erkennen von Aktivitäten von Personen anhand von verschiedenen Sensordaten beschäftigt. Dabei gibt es verschiedenen Applikationen, die von dieser Art von Informationen Gebrauch machen. Fittnesstracker können zum Beispiel benutzt werden, um zu bestimmen wann eine Person läuft oder mit dem Fahrrad fährt. Auch im Medizinsektor beziehungsweise in der Pflege kann Aktivitätserkennung helfen. Patienten mit kognitiven Einschränkungen könnte beispielsweise geholfen werden, wenn erkannt wurde, dass sie ihre tägliche Medizin nicht eingenommen haben. Weiterhin können Unternehmen mit Aktivitätserkennung Prozesse in der Logistik optimieren, etwa im Warenhausbetrieb. Dabei werden oft ein oder mehrere Sensoren für die Erkennung genutzt. Diese Arbeit beschäftigt sich mit dem Thema der multimodalen Aktivitätserkennung. Das heißt, dass wir mehrere Sensoren verschiedener Art für unsere Modelle nutzen. Dabei ist unsere Annahme, dass verschiedenen Modalitäten die Gesamtgenauigkeit von Modellen erhöhen. Da Sensoren in den letzten Jahren im Preis gesunken und gleichzeitig auch kleiner geworden sind, können multimodale Systeme auch leichter in der Praxis angewendet werden. In dieser Arbeit beschäftigen wir uns im Speziellen mit Video- und Beschleunigungssensoren. Hierbei haben wir Beiträge für jeden Schritt in einer typischen Aktivitätserkennungspipeline geleistet. Dazu gehört das Erstellen von neuen Datensätzen, die Entwicklung vereinfachter Annotationsmethoden und das Erarbeiten und Evaluieren von neuen Methoden für Aktivitätserkennung sowohl im Industriekontext als auch für persönliche Nutzung.

The question of whether machines man think is about as relevant as the question of whether submarines can swim.

— Edsger Wybe Dijkstra

ACKNOWLEDGEMENTS

Finishing a PhD is no easy feat. It comes with a lot of challenges: coming up with topics, setting up experiments, debugging code, writing the paper, waiting for reviews, and finally presenting the work. And then the process starts anew. Along the way, I faced multiple challenges. Rejected work, ideas not working out as planned, tough deadlines, and many more. So I would like to acknowledge people that helped me on the way to finish my work. First of all my adviser Heiner Stuckenschmidt. Without him I would often be stuck on a single task, burying myself in details that may not be as important in the long run. He would always help me see the bigger picture and find new angles on ideas, I had not considered before or even given up on. Also, I'd like to thank my second adviser Margret Keuper for sitting down with me and talking through ideas, giving me valuable input. Then, I would like to thank my colleagues Lydia Weiland and Timo Sztyler. Both of them helped me get started with my work as a researcher, showed me how to write papers, and were aiding me when I had questions. I would also like to thank my colleague Manuel Fink. Sharing an office with me, he was always available to talk over research ideas and, occasionally (or sometimes frequently), would just goof off with me whenever frustration levels were too high. Finally, I would like to thank my family. Namely, my parents, who always supported me in my endeavors and my sister and my brother in law, who always helped me to calm down in stressful times.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACRONYMS

IMU Inertial Measurement Unit

CMU MMAC Carnegie Mellon University Multi-Modal Activity Database

ADL Activities of Daily Living

EEG Electroencephalography

FFT Fast Fourier Transformation

MEMS Micro Electronic Mechanical System

SVM Support Vector Machine

RF Random Forest

ANN Artificial Neural Network

HMM Hidden Markov Model

PCA Principal Component Analysis

IoT Internet of Things

Part I

PRELIMINARIES

1 INTRODUCTION

1.1 PROBLEM SETTING

The recognition of human activities is a broad task that may overlap with many other fields. Behavioral science, medicine, and logistics are just a few examples where an automatic detection of human activities can be very beneficial. In this thesis, we explore different methods for detecting human activities based on different input information. Recent years have shown a great increase in sensor devices, may they be personal like smartphones, or ubiquitous sensors in our environments like cameras or motion detectors. With the increase in sensor availability, new possibilities for detecting human activities are opening up for researchers and companies. Use cases can be divided into different levels. To get an overview we try to describe the most common applications in the following subsections.

1.1.1 *Health sector*

For the health sector, activity recognition may be useful when considering care for patients [64] [93]. Given the fact that many nations are experiencing an aging society [14][68], personal care for elderly people for instance may not be feasible. A recognition system may be helpful here, as it enables care givers to validate whether a person is still able to perform everyday tasks by themselves. For this purpose, scholars in the field of medicine have defined so-called *activities of daily living* [53]. With these activities in mind, researchers have a way to test their methods in a meaningful way [80][8]. In many cases, health care approaches rely on smart home environments to detect activities. One important reason is the privacy concern for the patient. Especially in the health sector, lost data due to a user interfering is very problematic. Personal hygiene for example is part of the activities in ADL where patients may feel uncomfortable using personal sensors and therefore are likely to put them away. Smart environments can be built fairly non-intrusively if sensors like cameras are used sparsely.

1.1.2 *Industry*

In industrial settings, activity recognition may help to streamline processes for human workers [33][39]. One example is order picking in warehouse scenarios. Workers often have to pick items from huge rows of shelves and drop them off in carts. With devices like smartglasses being more broadly available, new possibilities to streamline this process emerge. Especially when combining activity recogni-

tion with the emerging field of augmented reality and smart environments, tasks like finding the right shelf or scanning picked products can be improved for the worker [55][37].

1.1.3 *Personal use*

Often, people use smart devices like their phones or watches to track their own activities. A typical use case is sports tracking. Big markets have been created for fitness trackers that are able to recognize how long and where a person is running, how many flights of stairs they climbed, and similar activities. Users nowadays have multiple methods to track vital information about their life without investing heavily in custom hardware. For these kinds of use cases, devices often use a combination of sensors to achieve high precision results. One of the most prominent examples is the fusion of acceleration, gyration, and GPS data for determining position and locomotion of a user [97]. From this data, fairly accurate life log information can be inferred. In fact, inertial-based navigation is even used in other fields like ships, aircraft, and autonomous robots. GPS information for instance is used to infer the position of a person at a given time, while the inertial data gives programs notion about the mode of transportation (i.e. walking, running, driving a car). The same sensors can also be fused to then decide if a person is currently moving by foot or if they use another form of transportation like a bike or a car. Here, the GPS data can give broad information about a user's speed with finer details inferred from inertial data.

1.2 APPROACHES

As briefly mentioned above, approaches in activity recognition can be divided into two categories. One approach uses sensors in a user's environment to infer an activity. The other utilizes sensors that a person carries with them to do the same. In the next subsection, we try to distinguish both categories and show their unique properties.

1.2.1 *Personal sensors*

Utilizing sensors on a user as means to detect activities has gained a lot of traction in recent years [7]. Devices like smartphones include many sensors that can be used to infer an activity. A big focus is put on inertial measurement units that often contain acceleration, gyration, and magnetic field sensors on one side, and GPS sensors on the other side. It is not uncommon to combine both of these sensors to get better results overall, especially in consumer applications like point-to-point navigation or fitness tracking which often try to establish the mode of transportation on top of locomotion information. Typical properties of personal sensor-based activities include:

CONSTANT STREAMING

Most approaches that utilize personal sensors are based on constant streams of sensor data. Given a fixed sampling rate, methods in this area try to learn patterns from these streams. Trying to detect a user walking for instance can be realized by isolating the typical walking pattern that can be seen in acceleration data streams. With each step, the accelerometer may detect the motion of the user first lifting their foot, then moving forward, and finally stepping on the floor.

LACK OF CONTEXT

Typically, approaches that utilize inertial measurement units do so without having any notion about a user's context. Applications thus offer predictions on lower levels e.g. running and walking but yield little information about why a user performed these activities.

COMPARABLY LOW COST

With an increasing trend towards smart devices [11][5], the cost for activity recognition systems may also shrink, as sensors become more available for a greater set of users. Additionally, custom built solutions (e.g. smart bracelets) are also cheaper to design, as sensors and microcontrollers are more widely available and easier to use (e.g. with the Arduino platform [59]) in addition to easier access to 3D-printing hardware.

We can see that personal sensors have their advantages when used for activity recognition. However, we also have to consider the limitations of this approach which include:

NEED FOR PROCESSING

For many use cases, the raw acceleration data is not feasible for activity recognition tasks. It can contain gravity information for example, which may influence classification results and therefore often needs to be subtracted from the data. Other popular methods utilize a windowing approach to abstract from single measurements and rather look at fixed time spans to recognize patterns. Overall, this leads to a certain amount of pre-processing for personal sensor data that has to occur before meaningful results can be achieved.

NOISE IN DATA

Little movements (e.g. when the sensor is loosely attached to a person) can already influence the returned measurements of sensors and add noise to the data. Especially with increased amounts of activities to be predicted, this noise can easily lead to wrong classification results. It is therefore an important part of any activity recognition pipeline to filter out or find another way to deal with any noise. Filtering noise can be done while collecting the data or after the fact e.g. with a threshold for captured data. However, this live filtering could theoretically impact the complexity of the collection process negatively, leading

to a lower sampling rate or missed data. Alternatively, feature extraction from the raw sensor data may also be able to deal with noise. Sliding windows would be an example of that.

DRIFT

Over time, measurements of inertial data can drift, meaning that values are becoming less precise the longer data is collected. Drift can occur due to aging electronic components within the sensor or outer influence like temperature or extreme vibrations. Another way this effect manifests is at a processing stage when the data is integrated (e.g to calculate positional information) and small errors accumulate over time. Regular calibration and sensor fusion are two methods that may help to minimize the drift effect.

POSITIONAL DEPENDENCY

Human activities involve movements of different parts of a user's body. Depending on which activity is supposed to be recognized, the placement of the sensor can be very crucial. Recognizing a person's steps, for example, might be easier with a sensor placed on the foot or shoe of a user compared to a wrist-worn sensor. This can lead to misclassification that users often experience in fitness trackers where independent arm movement is often classified as walking or running.

These limitations pose a set of challenges for researchers and may influence the feasible applications for personal sensor-based activity recognition.

1.2.2 *Smart environments*

Smart environments are another way how researchers approach activity recognition. The basic idea of smart environments (which can also be seen as part of the bigger community of ubiquitous computing) is to augment certain objects that surround a user with sensors in order to register interactions or usage. A popular example is the concept of a smart home, where rooms may be enhanced with presence sensors, electrical devices that measure current energy consumption, and sanitary installations that measure water usage to name a few. From all these interactions researchers may create a timeline of events that enables the inference of activities that a person has performed.

Let us consider the simple example in Figure 1. From the sequence of events, a human observer may easily conclude that the person was cooking food that they most likely took from the fridge. Even more so it is also likely that the dish being cooked needs boiling water as the water tap has been used before the stove has been turned on. We can see that given sufficient sensor coverage in an environment, many activities may be inferred just from simple events. From the example we can deduce some traits that smart environments possess:

1. `Fridge registers being opened`
2. `Cupboard with pans registers being opened`
3. `Water tap in the kitchen has been used for 30 seconds`
4. `Stove top registers that one of its burners has been turned on`
5. `The air filter over the stove top has been turned on`

Figure 1: A toy example of sensor events in a smart home

STATIONARY

Typically, sensors in a smart environment are stationary, which means that they do not change their location. This in turn makes it easier to reason about readings of the sensor, as its location and orientation are fixed in such a setting. Additionally, these types of information can make reasoning easier for specific information need. If we consider a stationary camera and compare it to a wearable one, we can clearly see that any type of movement captured in the frame is caused by a change in the environment and not by egomotion (unless external force is applied to the camera). On top of that, stationary sensors (e.g. presence sensors) guarantee that their readings are bound to the location of the sensor. Especially in multi-room scenarios, this background knowledge is very useful as it removes a certain level of uncertainty that personal sensors may contain.

MULTI-SENSOR

To get reasonable coverage of an environment, multiple sensors have to be deployed. This is aided by the fact that sensors in smart environments often serve only one specific purpose (e.g. a sensor that detects if a fridge was opened). Wearable sensors on the other hand may be used to infer multiple activities.

EVENT DRIVEN

Typically, activity recognition systems based on smart environments are event-driven. As many sensors are mostly putting out streams of data, minor amounts of processing of the data are needed before models can be applied. An example of this could be a presence detector that uses an ultrasonic sensor. Here, it would make sense to only output an event when something blocks the sensor's path below a certain threshold (and when the sensor readings are back above the threshold after some time).

DOMAIN SPECIFIC

Setups for smart homes have to be adapted to the environment where they are to be deployed most of the time. Since room layouts, home appliances, dimensions, and other parameters of peoples' living environments vary greatly between different

users, activity recognition systems may have to be adapted on a case-by-case basis. To give an example, we can consider a motion detection sensor that is used to track the location of a person by registering when they move from one room to another (e.g. by using light barriers in doorframes). In the easiest scenario, only the background knowledge about how each of the rooms is connected and where each specific doorframe is located has to be changed. But scenarios can get more complex, i.e. by considering houses that have a shared living room and kitchen area. Therefore, smart home systems for activity recognition often have to be custom tailored to the specific living environment.

Smart environments can offer good recognition rates for specific scenarios. But they also may have some drawbacks which are described below.

MULTI-USER

Recognizing activities of a single person living in such an environment may yield good results. However, when more people are living in the same space, the task may get significantly harder. A system then has to determine for each event who caused it, in order to still contain a cohesive sequence of activities for each user. In the worst-case scenario, a system attributes the events in such a manner that the behavior of one person may seem erratic and atypical. Since such systems may often be targeted for elderly care uses, such misclassifications are problematic. Having a unique way to identify a person (e.g. using an RFID tag) could help to mitigate the problem but leads to other issues (limited range, some sensors not working with tags, adding another point of failure). Another way to address these issues would be to increase the number of sensors, thus having a higher density of events and making it easier to attribute events to single users. This, however, leads to another downside of smart environments.

HIGHER COST

While it is true that common sensors are readily available and cheap, the fact that smart environments are often very domain-specific makes them more expensive than personal sensors. We can estimate that most of the cost would be in the configuration and installation of such a system, especially when sensors should be non-intrusive, thus hidden from view. On top of that, maintenance could be another issue as sensors may fail over time and have to be repaired or replaced. Looking at the software, updates in the model may have to be adapted for each environment individually while personal sensor models could ship updates independently.

PRIVACY

Security and privacy in the field of activity recognition is a general issue that researchers have to address (see Chapter 2). In

smart environments, this is especially prominent. With personal sensors, users may have the feeling they can put away the device and are thus no longer being monitored. Smart environments, on the other hand, are permanently attached to the surroundings of a user. Thus, even if the system is easy to shut down, the physical presence of sensors could still be a big concern for many users. This is especially true when these environments use cameras as sensors.

It can be seen that the field of activity recognition is fairly big and contains multiple sub-fields that have unique properties and features. In this work, our focus is on activity recognition with personal sensors. We want to look at combinations of sensors, typically in smart devices, that are carried by a user in order to predict activities.

With this work, we contributed in multiple ways to the field of activity recognition. For the first few steps of typical activity recognition pipelines, we created a new multimodal dataset. Multimodal datasets do exist, but often miss the crucial sensors or deal with unrelated scenarios for our use case. We wanted to analyze activities that are hard to distinguish based on their motion alone (e.g. food consumption versus medicine intake). Thus, our dataset contains a subset of a typical ADL scenario with the subjects performing very similar activities in different human body poses. Then, we created and tested a new method to speed up annotation processes. The annotation of multimodal data can be fairly time-consuming, especially when the annotator does not know when activities occur and has to watch the entire video. Therefore, we want to make use of the multimodality of the data to speed up the process. By utilizing already labeled data, we can make labeling recommendations for the annotator, speeding up the whole annotation process. After data collection and annotation we can look at the core problem which is the correct recognition of activities. We looked at two problems: activity recognition in an industrial setting, specifically warehouse logistics, and scenarios that target everyday life like cooking or activities of daily living. For both scenarios, we considered egocentric vision and inertial information as our input data, whereby we utilized HoG, object detection, and sliding windows as our feature generation methods. We developed two approaches and evaluated them, being able to show improvements over existing solutions.

2

FOUNDATIONS

2.1 ACTIVITY RECOGNITION

In this chapter, we explore the definitions of activity recognition and give an introduction to the main sensors used in our work. First, we define different granularity levels for activities in order to have better categories for our algorithms in the later chapters. Afterward, we consider two types of activities in more detail: cyclic and complex activities. We then take a closer look at the features of multimodality. Finally, we examine inertial sensors in more detail as they are one of the main component in our work.

2.1.1 *Granularity*

When we talk about human activities, levels of granularity are important [16][52]. Without such a distinction we would group together long activities that are semantically rich with small movements of a person that are hard to place into context. Therefore, a separation into three classes has been proposed, which can be seen in Figure 2. We can further describe the different levels as follows:

GESTURE

Gestures are the smallest unit to define an activity. In extreme cases, they can be static poses of a human, for instance when showing a hand sign. Gestures are often used to interact with software systems to quickly access functionality. One example is camera applications on phones that take pictures if a person shows a specific gesture (e.g. thumbs-up sign).

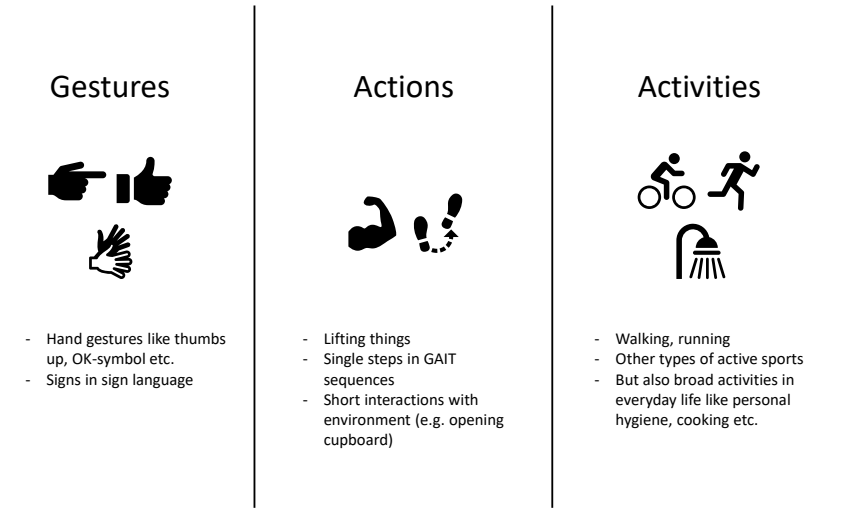


Figure 2: Different levels of granularity that researchers try to detect.

ACTION

Actions can be made up of multiple gestures. They are small movements that usually do not contain too much information about the whole activity a person is performing. Since they are positioned at a medium granularity, they are used for a broad range of applications. Similar to gestures, they can be used to interact with applications. Smartwatches or bracelets can, for instance, detect the arm and wrist rotation of its wearer to turn on the display and show the time.

ACTIVITY

Activities can be made up of multiple actions and are the most semantically rich of the three. In most cases, this is the level of granularity that we want to predict in this work. Examples of activities that are often targeted are locomotive activities like walking and running or everyday tasks that people perform in their homes like eating, sleeping, and watching TV.

While the definition makes big distinctions among the three levels of granularity, in practice it is often difficult to assign them to a given sequence. Let us consider the example of a person opening a cupboard. The whole sequence is semantically clear and can therefore be seen as an activity. On the motion side of things, however, opening a cupboard only consists of raising one's arm, grabbing a handle, and pulling. Therefore one could argue that this is an action rather than an activity.

2.1.2 *Cyclic Activities*

Many activities that are being researched can be considered cyclic in nature. This means that they are typically repeating themselves for longer periods of time. Examples for cyclic activities include walking, running, or climbing stairs. These specific activities also contain the **Gait cycle**, which describes the different single motions of a person's feet when taking one step.

2.1.3 *Complex Activities*

In this work we consider complex activities to be the main focus of our research. We define complex activities to be made up of a set of actions that are also dependent on context. Typical examples we use in this work come from the cooking domain. To cook a recipe, a set of steps is needed and activities performed in this case can depend on the context. Opening different packaged items of food, for example, can involve different actions depending on the food's packaging.

2.2 MULTIMODALITY

For many of the activities we want to consider, using only one type of sensor may not be feasible. We can distinguish between two main ways in which different sensors are used in a multimodal setting.

ADDITIVE SENSORS
: In this case, we may take into account different sensors to give us information about the same mode we are interested in. To give an example we can consider movement detection with two sensors. One of them is a first-person video feed, the other an inertial sensor. Both sensors can be used to detect the movement of the person wearing them. However, by combining the readings we may get more robust results. Video in this instance may have trouble dealing with movement that occurred in the wearer's environment, leading to misclassification. Inertial sensors on the other hand may pick up small movements (e.g. swinging an arm) that do not belong to the target class. The combination of both sensors may reduce these ambiguities.

COMPLEMENTARY MODALITIES
: Another approach is using complementary modalities to make predictions. This can often be the case when activities occur in different modalities. For another example, we can once again consider both a video and an inertial sensor. Inertial data gives us information about the movement of a person. With a certain error rate we can therefore register distinct actions. Video in this context, however, may give us information about the environment. One such piece of information is the objects that a person is using or interacting with. With the joint data, we may have an overall notion about what activities (especially ones that involve objects) a person performs.

Often, one can argue that a hybrid approach between the additive and the complementary approach is useful. In the context of the aforementioned example, video, and inertial data can on one side be considered both for movement and object information respectively while on the other side video data can enhance the detection of movement (e.g. when arm movement is also visible in the video frame).

2.2.1 *Alignment*

Data from sensors that measure with a fixed frequency has a set sampling rate and thus runs on its own clock. This poses a problem when we want to examine occurrences at a certain point in time among a set of disconnected sensors. Different sensors typically run on separate clocks and with different sampling rates, thus having few or no data points occurring at the exact same time. Therefore, we face an issue where we do not have a notion about temporal relations between different sensor readings. This problem can be addressed either while recording the data or afterward as part of the pre-processing step

in our activity recognition pipeline. When we try to solve the issue while recording, one typical method is using a central recording node e.g. a computer that is connected with all the sensors. This node may be running on its own clock as well as getting timestamp information from all other sensors to build a mapping between the sensor's clock and the central clock of the recording node. With the mapping in place, incoming data can be buffered and then resampled to fit one time series. Here we would use methods like down- and upsampling as well as interpolation to solve the issue.

When data is recorded independently, alignment can be more difficult. One big problem that arises, in that case, is the lack of knowledge about the time difference between multiple sensors. This is often solved with an action that is captured on all sensors and that can be identified easily. An example would be to record a zeroline (a phase where no action occurs) and then have a subject perform a sudden movement that is easily observable in every sensor. Another solution would be to start the recording of all sensors at the exact same time which is often unfeasible without a more complicated sensor network architecture (e.g. like the aforementioned central recording node).

2.2.2 *Fusion*

Fusion is the step in the activity recognition pipeline where we have to combine the data from different sensors to get an overall classification result. Generally, we can differentiate between early and late fusion methods. With early fusion, we take the data (usually after feature generation) and fuse it before we classify the results. Late fusion, on the other hand, is treating each sensor (or group of sensors) independently, classifying them on their own. Afterward, the results are fused for one final result. We may also see that fusion methods can be dependent on the sensor types and whether they are additive or complementary modalities. Additive sensors are a good candidate for early fusion since they are concerned with capturing the same type of information. Thus, a model may profit from having all data about one mode available at the same time to make a unified decision. Complementary sensors, on the other hand, can arguably be a suitable use case for late fusion. We can assume that separate models are learned independently for each mode that has been captured without getting confused about the other modes. The combination of both models can then be seen as a good indicator for the activity that has been performed.

2.2.3 *Security and Privacy*

Activity recognition in general faces the challenge of privacy as briefly mentioned in the Introduction. Users are often concerned with the amount and type of data that is collected about them. Wearable devices like smart bracelets, for example, may collect a lot of data that

can be deemed sensitive as it relates to the health of the user. This may include the heart rate, steps per day, or even oxygen levels in the bloodstream. While this information is vital for the intended purpose of measuring fitness, a sudden leak of this information could have dire consequences for users. In the scenarios we consider, this problem is even more pronounced as we also utilize the modality of video. Here, users are especially attentive regarding their privacy since video sensors are typically easier understood than other devices. Additionally, they may be an even bigger risk as they can gather information about the surroundings of a user without the consent of others.

We can see that privacy is a big concern when considering activity recognition. To deal with this issue, we can look at two principles that can be applied to activity recognition to preserve privacy. One principle is to minimize the amount of data that is collected on a person. If a system is able to work sufficiently with only one sensor, then this should be the only one used. An overabundance of collected information may lead to information leakage which breaches the privacy of a user. The other big principle is building data processing pipelines that work offline. When the user can be assured that the activity recognition works only on their device and never leaves it, then they have (to an extent) physical ownership of their data. Thus, they can be sure that their information stays only with them.

2.3 SENSORS

Since this work mainly deals with inertial data, we have to define the single components within these sensors. We also briefly touch upon typical types of sensors that are combined within inertial sensors.

2.3.1 *Inertial sensors*

A major point of this work is the usage of inertial sensors. We especially look at accelerometer data as an indicator for activities. In this part, we define what inertial sensors are and give details about the typical modalities built into them.

When we work with inertial sensors, we typically talk about inertial measurement units (IMU). These IMUs consist of accelerometers and gyroscopes with many of them also containing magnetometers. Often, sensors are labeled with a certain amount of degrees of freedom (DOF). This number represents how many independent readings a sensor can provide. So an IMU with an accelerometer and a gyroscope may have six degrees of freedom as each sensor provides three independent measurements per reading. Figure 3 shows a MEMS inertial sensor that provides acceleration, gyration and magnetic field data. Such sensors are cheaply available and way smaller versions

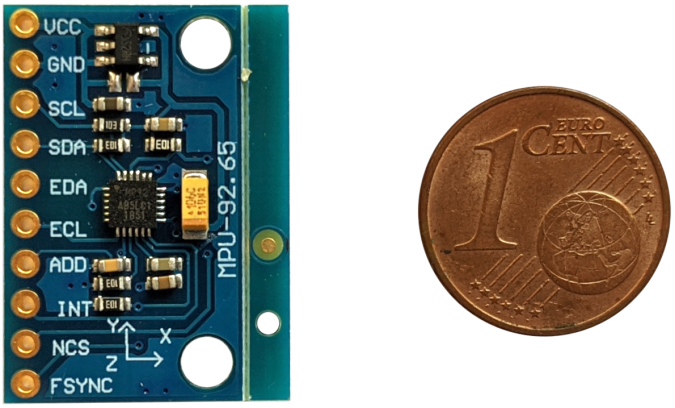


Figure 3: A 9-DOF inertial measurement unit for DIY electronics. These types of sensors have become cheap and readily available for hobbyists allowing for widespread usage in different projects.

are built into many consumer electronics like smartphones and smartwatches.

2.3.2 *Accelerometer*

Accelerometers are sensors that can measure acceleration. The information is usually given with three values, each representing acceleration for one separate axis. Insight about acceleration can only be provided relative to the sensor's position and orientation. This, in turn, makes the sensor's placement an important aspect for activity recognition, as we do have to consider the sensor's information based on its current location and orientation.

Internally, there are multiple ways accelerometers can measure acceleration. A very common type is the piezoelectric accelerometer. This works with a small crystal that can move freely in an enclosure. When force is applied to the sensor, the crystal hits the wall of its enclosure. This in turn induces a small current that can be measured and then converted to acceleration values. Figure 4 shows an example plot of raw acceleration data. Here we can also see the effect of gravity regarding the measured values. The acceleration along the y-axis is always around $10\mathrm{m}/\mathrm{s}^2$ which roughly corresponds to the earth's gravity.

2.3.3 *Gyroscope*

From the information about acceleration, we can already deduce a certain amount of information. However, we run into issues when we consider typical human movements. These are almost exclusively not rigid motions along the three relative axes of the sensor. Rather, human motion often also involves rotational movements. This is where

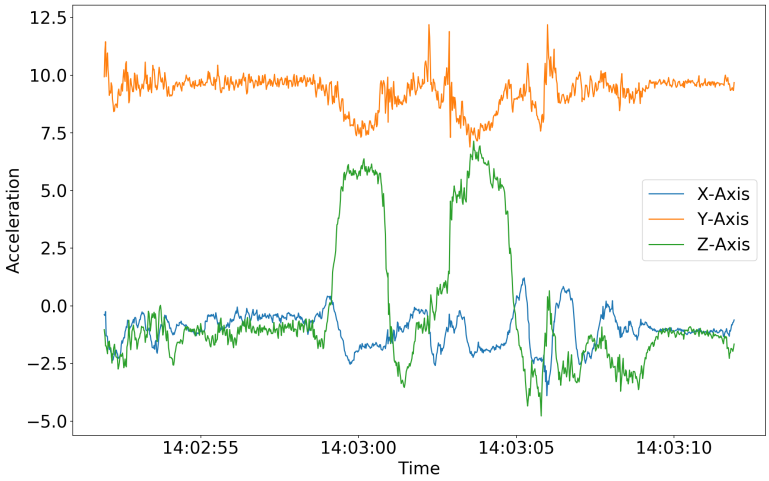


Figure 4: Raw acceleration data, taken from a smartphone with a sampling rate of 50Hz.

the gyroscope gives us vital information.

A gyroscope typically measures the relative angular velocity of the sensor, typically in degrees per second. Vibrating structure gyroscopes are often used in consumer electronics as they are typically cheap to produce. They work via a constantly vibrating structure within the sensor that is contained in one plane. If a rotational force is applied to the sensor, the Coriolis effect occurs and affects the structure. This way, the structure is putting pressure on its frame, thus inducing a current which is measured and translated to rotational angle. In order to get all three degrees of freedom, multiple vibrating structures are often used in a gyroscope.

When using gyroscopes, especially for locomotion purposes, we have to consider the bias of the sensor. Intuitively, this is defined as the output the sensor produces when no rotation occurs. If this value is not considered in calculations, the error of the bias is propagated to subsequent calculations and accumulates very quickly. In order to accommodate for the bias, a recording of the output of the sensor over a longer period of time is made. From this output, an average can be calculated to estimate the bias. This average can be subtracted for each reading to get more precise results. However, the bias can change over time, so frequent recalibration has to be considered as well.

2.3.4 *Magnetometer*

One final modality that is often built into IMUs is a magnetometer. This sensor can provide information about magnetic fields. The simplest implementation of such a sensor would be a compass. Magnetic field data is useful, as it provides a method to relate the sensor's ori-

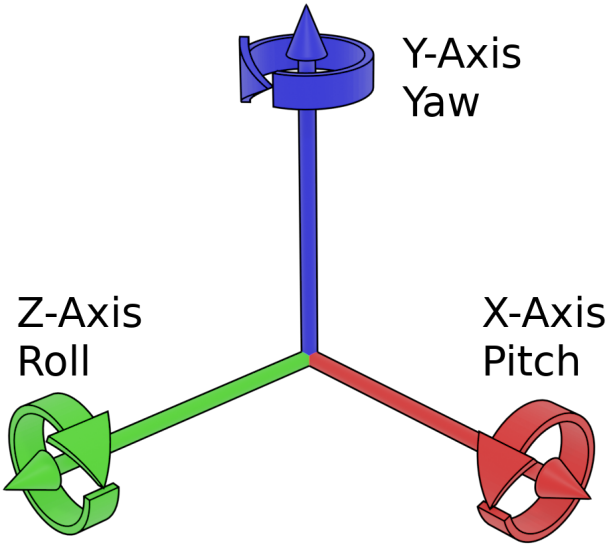


Figure 5: Three axes in three-dimensional space. Rotational movement is often given via yaw, pitch, and roll values. Typically, only pitch and roll can be calculated with accelerometer and gyroscope data.

entation to its environment.

In phones, the magnetic field information is typically measured via linear Hall sensors. It works with a strip of metal that has a current applied to it. When a magnetic field is applied, electrons in the strip are deflected to one side as they are affected by the Lorentz force. With the electrons distributed over two sides of the strip, a current can be read between both ends. Using multiple of these strips in different orientations, the location of the earth's magnetic field can be determined.

Magnetometers have issues working properly in indoor environments. They pick up on other objects made from metal in their surroundings, which adds errors to their readings. Additionally, the measurable strength of the earth's magnetic field is dependent on the location of the sensor on the planet. Thus, some information derived from the sensor can only be used in conjunction with GPS sensors.

2.3.5 *Calculating meaningful features*

We have seen that inertial measurement units can measure acceleration, angular velocity, and even magnetic field information. A typical use case for these types of information is calculating the orientation and position of the sensor [97]. With data from the accelerometer and the gyroscope, relative position and orientation (called *yaw*, *pitch*, and *roll*, see Figure 5) can be inferred, though *yaw* can only be estimated given an assumed initial value. Intuitively, we first assume (or get to know via user input) the initial position of the sensor. Over time the acceleration and angular velocity are measured. If we integrate

the values, we can infer the position and orientation of the sensor. In practice, more elaborate sensor fusion methods are used to get more precise values as sensors suffer from noise and drift which results in less accurate readings. Here, the disadvantages of both sensors can be negated by the corresponding other sensor. A gyroscope may be more prone to drift while the accelerometer may be affected more by noise. Fusing both values e.g. by first using high-pass and low-pass filters to remove the noise and drift could provide us more stable results. Magnetic field information can give us even more insight. From its values *yaw* can be calculated, yielding even more details. This is due to the fact that we have both a notion about the earth's magnetic field and it's gravity, both relative to the sensor.

The previously mentioned features are typically used for tasks like tracking in different fields. In the case of activity recognition, other features may also be used for better detection. Oftentimes, these are less intuitive than the orientation of the sensor and are based on windows of data. Windows allow us to consider a small subset of values at a time and calculate features that represent the whole subset. In this work, we can distinguish between features in the time and in the frequency domain. Raw data is given in the time domain with a fixed sampling rate for the values. To transform it into the frequency domain we use Fourier transformation (typically Fast Fourier Transformation - *FFT*).

Part II

CONTRIBUTIONS

3 ANNOTATION OF ACTIVITY DATA

3.1 INTRODUCTION

For a huge set of tasks, especially in activity recognition, researchers need labeled data. A special case is the field of multimodality (e.g. the combination of inertial data and egocentric vision [24]), as annotations are usually carried over to other modalities. Proper annotation of data can be time-consuming due to the amount of data and, in some cases, because of the need for knowledge of domain experts. What usually takes up most of the time is watching the full length of video sequences as it may be the case that annotators do not have previous knowledge about the activities performed in the data. Without such knowledge, it is necessary to watch the videos in their entirety or otherwise risk missing activities. An automated annotation approach is thus favorable and recent developments in wearable devices [77] may just enable such solutions. Every additional device that records data offers more opportunities for annotation tools to make their predictions.

Since most machine learning algorithms rely on training examples to learn their predictions, manual annotation is still a factor in these scenarios.

Therefore, for models that assist users in annotating data to be useful, we are bound to annotate data without any aid until a satisfactory prediction rate of the algorithms is achieved. Other approaches for annotating data as presented in [36, 48] rely on visual support to help annotators in their efforts. While this is a huge help, automatic label recommendations would be more beneficial, as these would speed up the process even more. Visual feedback, however, may enable annotators to see patterns in the data, for example how specific movements are represented in inertial data when plotted. Based on these patterns, annotators may jump within the timeline to quickly label big subsets of the data. We can see that annotators somehow *learn* the patterns which is a process we try to mirror in our machine learning approaches.

To tackle the task of annotating multi-sensor data, specifically in the field of activity recognition, we developed an application that can start to aid annotators after it has been fed a relatively small amount of labeled examples. The tool will then provide annotation suggestions that the user can confirm, reject, or modify, thus speeding up the whole annotation process within a dataset containing similar activities. In this research, we focus on the method and its performance in terms of correct annotation suggestions rather than on usability of the software or overall efficiency of the implementation. We focus on finding as many activities as possible at the cost of precision of

our predictions since we believe that it is easier to discard or adjust a label that was assigned a few seconds off the correct position instead of finding missing annotations. On top of that, annotation of data in itself can be very user-dependent, as agreement on the start and end of activities varies among different annotators. Thus, it is sensible to offer more annotations in order to make the work easier for as many annotators as possible. Initially, we developed a web-based application that provides support concerning the alignment, analysis, and labeling of inertial and video data. This web application plots acceleration data and shows the video of the recording at the same time. After alignment has been set by the user, both the plot and the video are synchronized bi-directionally, such that interaction with one of them (i.e. scrubbing in the timeline) updates the other visualization and vise versa. Then the user can annotate data within the website. However, web technologies are not flexible enough to offer the amount of control we need over video feeds, thus not allowing us to be as precise as we want. We, therefore, rewrote the tool to a native application using OpenCV [15] and python TK which also allows us to implement our label suggestion method. With the visualization and recommendations, the task of labeling may also be assigned to non-domain experts.

We test our system on different datasets with different activities, ranging from simple actions like grabbing items from a shelf to complex activities like preparing a whole meal. For the short and simple activities, we look at an industry dataset that is concerned with order picking in warehouses (henceforth called **Picking**). To clarify: picking in this context means the selection of items from boxes in shelves of warehouses that make up an order, e.g. for a customer.

Data was gathered with different devices: we use a custom wristband and smartglasses which collect inertial data and egocentric video respectively. Of the inertial data we collected with the band, we focus on acceleration in our experiments.

For a more complex scenario, we try to recognize **Activities of Daily Living** from a dataset we created ourselves. It again contains egocentric video from smartglasses and inertial data collected by smartwatches on both wrists. The focus of the dataset is the recognition of hard to distinguish activities. These we define as activities that consist of similar motions with the arms. Specifically, in our dataset, these revolve around consuming things like water, food, or medicine. As our method exploits characteristics of motions, we try to recognize all activities at once without differentiating between the specific motion. Furthermore, we focus on the subset of activities that were performed while lying down. This allows us to further broaden our experiments, as the other activities in the different datasets were all performed while standing up. Henceforth this dataset is called **ADL**.

Finally, we look at complex activities involving multiple movements that form an activity. Here we show that our method is able to cluster similar activities based on their motion. We use the CMU-MMAC [24]

dataset (Carnegie Mellon University Multimodal Activity) for our analysis as the cooking activities present are full of multiple movements per activity. In the following parts, we call this dataset **Kitchen**.

In this part of the work, we explore the matching of acceleration data to automatically annotate datasets, specifically in multimodal scenarios. For that purpose, we focus on template matching in the form of dynamic time warping as preceding works already presented promising results [61]. We are focusing on the question of how well a template matching based approach can be used to recommend labels and an analysis on which methods work well for different datasets. Our contribution is the application of dynamic time warping for recognizing activities among a broad spectrum of data with a focus on supporting manual annotation of data while also supplying an in-depth analysis of factors that influence the results.

This chapter reflects the content of our previous publications [30][31].

3.2 RELATED WORK

Annotation of activities and the quality of the labels is very much dependent on the tools used to annotate data. This was investigated by Szewcyzk et al. [86] who were able to show that with increasing assistance (in the form of visualization and predefined activities) annotators perform a labeling task with higher accuracy and in less time. Therefore different methods of annotating data (especially videos) have been researched [26, 27, 58]. Clustering of video information and subsequent visual representation in form of a multi-color navigation was shown to improve the annotation task [27] while methods of browsing videos non-sequentially and in parallel let users grasp the content of a video faster [26]. In many respects though, especially the automatic annotation of video data is challenging [45]. Therefore many different approaches in automatic annotation have been proposed in previous publications. These are either purely vision-based or based on sensor data.

A pure, vision-based approach was presented by D'Orazio et al. [20] who were able to improve video annotation for soccer games by first applying a pre-trained model to recognize soccer players and afterward having the annotators correct the misclassified positions. This approach is hard to apply to our scenario though, as the scenarios we consider are located in a more open-world setting. For this approach to be feasible, we would need classifiers for a lot of different objects and backgrounds. This might still not cover all activities, as egocentric video often has the issue of activities only being partially in frame. In addition, our activities are not defined by object occurrence in a frame but rather by the interaction of the user with objects or their environment in general. For that purpose, we focus on the feasibility of transferring automatically recognized labels of acceleration sensor data to corresponding video recordings. Therefore, we

focus on template extraction and matching of certain motions.

Such a template-based approach was suggested similarly by Margarito et al. [61] who already showed how templates that are extracted from a wrist-worn accelerometer sensor are able to recognize certain sports activities across different people. Furthermore, they pointed out that combining different template-matching metrics in the context of statistical classifiers could also be promising. Similarly, Martindale et al. [62] used Hidden Markov Models to find annotations and showed good performances for cyclic activities. While the results of these works are promising, the activities that were considered were all cyclic in nature, like walking, cycling, and squatting. In contrast, we try to find few, mostly very short activities in recordings with a similar length where we cannot rely on the cyclic property of the activities.

Besides template matching, Spriggs et al. [84] investigated a multimodal based classification approach considering inertial sensors but also adding first-person video data. They focused on daily kitchen activities and performed a frame-based classification by relying on features that were extracted from the inertial sensor and video data. However, they clearly state that their approach does not generalize well across people.

Relying only on inertial and force sensors, Morganti et al. [65] stated that inconsistencies as minor as different wrist shapes and muscle configurations across people can affect the recognition procedure. Furthermore, they point out that especially the force sensors they used in their custom wristband enable recognition of specific gestures that could not reliably be recognized by inertial sensors. While their approach is promising, the experiments they presented were in a preliminary state. Moreover, the types of sensors used in the approach are as of yet uncommon in off-the-shelf hardware and thus do not integrate with our scenarios.

Focusing on sensor data annotation tools, several researchers already presented powerful and promising approaches. But only a few of them provide support concerning labeling recommendations or automated labeling. Palotai et al. [72] presented a labeling framework that relies on common machine learning approaches, but was only designed for domain experts. In addition, it is unclear how their approach performs concerning different levels of activity types or how different sensors are supported with respect to their introduced learning approach (e.g. feature extraction). Indeed, Barz et al. [10] highlighted that most data acquisition and annotation tools are mostly limited to a particular sensor. This can be attributed to the fact that it seems to be necessary to consider different techniques or feature sets for different kinds of sensors. Especially the combination of multiple learned models in the context of automated labeling seems to be challenging.

It can be seen that for many scenarios, methods for annotating data are differing greatly. Some of them use image features, while others rely on inertial sensors. We, therefore, analyze one base method and adapt it for different datasets to see if we can achieve consistent performance among them. Since the datasets cover different scenarios, the adaption of the method is necessary as we even have slightly different tasks per dataset (e.g. finding a single label, finding multiple labels, etc.). For this purpose, we use dynamic time warping as it has shown to work on different types of sensor data [17, 61, 66].

To show an overview, we compare our presented approach to similar solutions and point out the differences:

OUR APPROACH

Main idea: Suggest labels based on a small subset of annotations. An in-depth analysis of different pre-processing methods and variants of dynamic time warping is provided.
Method: The main focus is the analysis of different variants of dynamic time warping used for label- and clustering-suggestions. The methods were applied to wrist-worn sensors mostly. Evaluation metrics used here were the time offset with respect to the correct label and the recall of the method.
Pros and cons: An in-depth analysis among different datasets with different configurations is given. Currently, the tool itself is just a prototype and therefore the usability of the tool is not tested.

LABEL MOVIE [72]

Main idea: Designing a complete multimedia annotation tool with automatic annotation and crowd-sourcing capabilities.
Method: Used dynamic time warping and SVM time series prediction with a focus on usability of the application. Classification results are shown in a Gram matrix to the user. Focused on crowd-sourcing capabilities with the combination of domain experts and technical expert's knowledge.
Pros and cons: The tool is fully developed with a lot of functionality, especially the capability for crowd-sourcing. On the downside, the evaluation of the tool is lacking in detail and it is not publicly available.

MULTIMODAL MULTISENSOR ACTIVITY ANNOTATION TOOL [10]

Main idea: A multimodal annotation tool that is able to handle multiple sensor types like video, depth, and body-worn sensors.
Method: The focus is put on capturing many different types of sensors and displaying them in a useful fashion. In contrast to the other methods, this tool is able to capture sensors live and synchronize them. Capabilities for automated annotation are present, but not implemented yet.
Pros and cons: Live capturing of different types of sensors is integrated and the tool seems to be designed very concisely. But

as of yet automatic annotations are not integrated though the architecture allows for that.

SMART VIDEO BROWSING [27]

Main idea: Using clustering methods, automatically segment videos into different parts to improve navigation within a video.
Method: For clustering, the tool uses color and motion features to distinguish different parts of the video. These can be browsed by the user to navigate the video faster.
Pros and cons: The tool does not rely on pre-trained methods and can thus easily be used. It does not, however, provide automatic labeling functionality.

3.3 DATASET

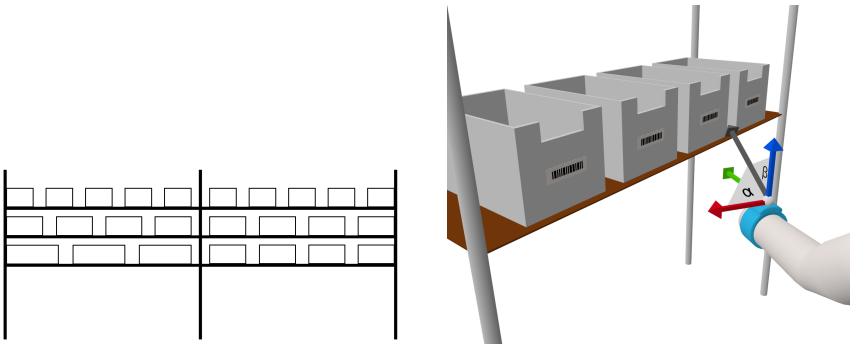
In order to test our approach on a broad spectrum of activities, we consider three different datasets which deal with different scenarios but use similar sensors. This allows for a broader analysis of our tool, as we are not limited to one specific kind of motion. The first dataset contains activities in the field of logistics (**picking** scenario). Here we analyze picking activities [23] that are used to collect items for an order in a warehouse. Each sequence contains one grabbing activity which is always performed while standing in front of a shelf. For the second dataset, we consider **activities of daily living** [53] and record a subset of these that specifically focus on activities with similar motions. Each sequence contains multiple activities and all of them were performed while lying down. The last dataset we use is the publicly available CMU-MMAC dataset [24] (**kitchen** dataset). This dataset is the most complex one as it involves a variety of activities with different motions that span longer periods of time.

3.3.1 *Picking scenario*

The data recording followed a predefined protocol that contains a sequence of activities, i.e., *walking to shelf*, *locating the correct box*, and *grabbing from the box*. In this context, several scenarios were recorded including picking from various boxes on different rows and from different shelves. These sequences represent an average picking job that a warehouse worker may have to perform in their daily work life.

The test environment consists of two shelves located next to each other where each shelf has three rows of boxes with three to five boxes per row. Additionally, the boxes were placed on different heights and were spread horizontally among two shelves (see Figure 6a). The test environment is set up based on real warehouses and tries to show a cutout of typical rows of shelves. A problem with recognizing grabbing motions is the variation of that activity, thus, a grabbing motion can produce highly different sensor outputs depending on the location of the object to be grabbed. In contrast, activities like walking or running typically do not have this degree of variation as movement patterns are often very regular in nature. Therefore, our

dataset contains multiple different cases of grabbing within the shelf to cover a bigger part of the space of different motions.



(a) Schematic of the shelves in our test environment

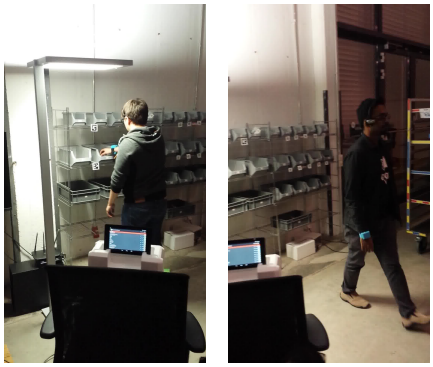
(b) Angle features from the wristband used for matching

Figure 6

The required data for our dataset was collected using smartglasses[1] and a custom wristband that consists of a 3D printed case with a wirelessly enabled inertial measurement unit and a battery. Both devices recorded acceleration, gyration, and magnetic field data while the smartglasses also recorded video information of their front camera. While the three inertial data modalities can be sampled synchronously on the wristband, the smartglasses are not able to do that. This is due to an android specific system design where sensor data is not queried by an application but rather pushed by the system which in turn gives no guarantee in regards to the specific sampling frequency. Further, as the wristband and the smartglasses are not connected with each other, the recorded timestamp of the data has to be synchronized in a processing step after recording. For that purpose, the subjects were instructed to stand idle for a some time before and after the performance of the activities to have a reference for alignment (see Method section). Data on the wristband was collected at 40 Hz for all the sensors while the smartglasses recorded the sensors at 50 Hz and 25 fps respectively. These values were chosen to give us the highest frequency possible without encountering performance issues, especially in the smartglasses. For better interpretation, each recording session was also filmed from a third-person perspective (see Figure 7). We use a depth-enabled camera on an Android tablet, which allows us to collect depth information of the recorded images in the form of point clouds.

For the recording, we relied on a self-developed application. Here, we enhanced an Android application of a previous work [87], where we specifically added the support for video and tuned the application for the use with smartglasses. The recorded data by the smartglasses is stored locally on the device. However, the custom wristband does not have enough storage to store the data locally; hence, we had to send the recorded data directly over Wi-Fi to a server.

1 Vuzix M100, Android 4.0



(a) Picking scenario. Participants walking towards and away from shelves.

(b) ADL scenario. A glass of water and pillbox on the table.

Figure 7: Photos from the recording processes of our datasets.

3.3.2 *Activities of daily living scenario*

We also recorded an additional dataset that contains activities of daily living. In total, we collected data of two participants who performed multiple activities per recording. Each one lasts between one and three minutes with a maximum of four activities per recording. This dataset focuses on activities that can be hard to distinguish due to similar motions. Specifically, we looked at food, water, and medicine consumption. All these activities include motions of reaching towards an item and then consuming it orally thus making them hard to distinguish just based on the inertial data. The data was collected similarly to the previous example but instead of using the custom wristband, we used a smartwatch (and a corresponding smartphone that was located in the test subject's pants pocket). In addition, we added another watch on the non-dominant hand, enabling us to capture motions from both wrists of the test subject. Also, we used the aforementioned tablet as a chest-mounted camera recording another perspective of egocentric video without collecting depth information. In total, we collected data from six devices: IMU from both phones, both watches, the tablet, and the smartglasses as well as video from the smartglasses and the tablet. We also recorded the whole test scenario from a third-person perspective to make annotating the data easier. For this work we looked at three different activities: *eating prepared food from a plate*, *taking medicine*, and *drinking water*.

3.3.3 *Kitchen scenario*

To add more types of activities, we also considered the *CMU-MMAC* dataset which was published and described in great detail by F. Torre et al. [24]. In contrast to our own datasets, this one is far more complex in multiple ways. It contains more different arm motions, e.g. getting a cup from a shelf includes opening the shelf and grabbing the cup. The setting of a kitchen leads to many different arm movements for retrieving objects. In consequence, the motions themselves

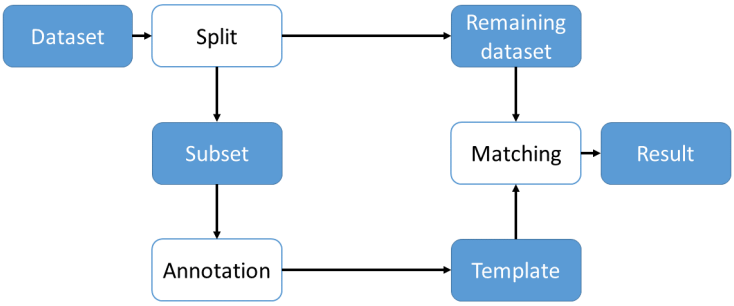


Figure 8: Basic approach for finding matches in a dataset. Blue boxes represent data, white boxes the processing of data.

in this dataset are not as homogeneous as they are in ours. Like in our activities of daily living scenario, they also recorded the movement of both arms instead of just one. But here we could not assume which hand was the dominant one for each participant confidently.

In our experiments, we consider both cases, i.e. simple and complex, to clarify the feasibility and performance of our approach. In this context, we consider a subset of the *CMU-MMAC* dataset. In particular, we only looked at one recipe, i.e. the brownie recipe, for a subset of all the participants because it was the only one that was completely labeled at that point in time.

3.4 METHOD

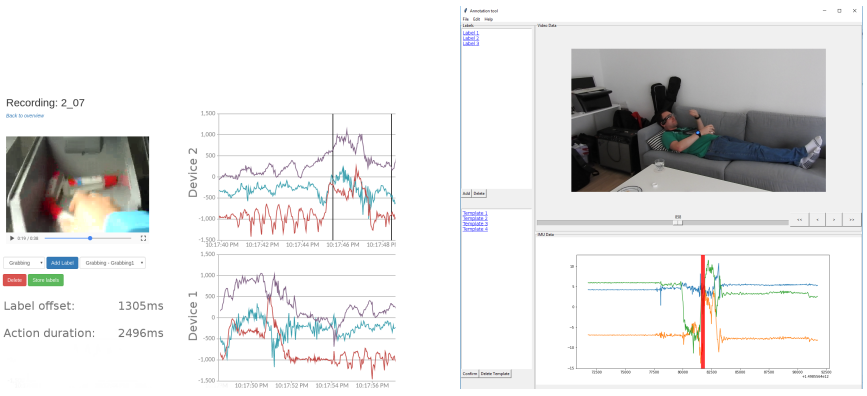
The method section is divided into three parts, each dealing with one of the aforementioned datasets. Since the datasets are differing in the type of activities, we are presenting separate methods for each dataset. We start with the picking dataset as this is the first scenario we considered, then move on to the ADL dataset. Here, we extend the methods to deal with the increase in devices and differing setup.s At the end, we present methods for the CMU-MMAC dataset which differ from the previous two approaches as they are more concerned with finding similarities among complex activities. All methods are dealing with our goal of offering label suggestions towards the user and do not contain aspects of usability within the annotation tool itself.

Figure 8 shows the basic approach used to find matches in a dataset. We label a subset of data and use these labels to create templates that are used for matching in the rest of the dataset. The approach for each dataset will be described in the following subsections. As we described in Section 3.2, methods for matching may have to be adapted when considering different data. Therefore, in an application, the specific method has to be chosen based on the data that has to be annotated. Regarding our own dataset, we also used similar approaches to align our data which was recorded with non-synchronized devices. Both datasets were recorded with a third-person camera for labeling purposes and contain distinct points in time that enable the alignment of the data. To align the data we first annotate the distinct

motion in the third-person video and, for the evaluation later, the labels we want to find. Then, we plot the acceleration data we want to analyze and find the same distinct motion in our data. Once both points in time are found, we can then map the labels from the video to the acceleration data and validate its position in time. For this purpose, we display the labels in time as an overlay of the acceleration plot and therefore can see if the labels are correct. As our datasets do not contain individual recordings that are longer than five minutes and sampling rates of sensors are stable, the drift of time and delay in transmission are negligible.

3.4.1 *Picking dataset*

As a first step, we align our picking data with respect to the timestamp. Here, we consider zerolines (a period of no movement) at the beginning of each recording, which allow us to pinpoint the starting time of one specific activity. More precisely, we use the first peaks of walking motions after the zerolines that are visible in accelerometer plots to align the data as those are easily identifiable as the first activity. This is in line with the process described in the introduction of this section, with walking being the distinct motion for alignment. We were considering an alignment pipeline for the tool but since alignment methods may vary among datasets, alignment information has to be created externally. After this step, we have consistent time information among all modalities. Subsequently, we label a grabbing activity, analyzing the acceleration sensor data that represents the motion and crosschecking against the video data that describes the same time period. This allows us to label all sensor recordings simultaneously. Once the boundaries of an activity are defined, the application replays the corresponding part of the video that was recorded with the smartglasses. After the confirmation of the correctness, the corresponding acceleration sensor data is extracted for creating a template of this activity where a template is represented by a start and end timestamp, the corresponding acceleration data, and a label.



(a) Old web application (b) New desktop application

Figure 9: Developed labeling tools. First version running in a browser, second version as a standalone application.

For now, we focus on the acceleration data because preliminary experiments have shown, that the angles relative to the three axes (Section 3.3, Figure 6) are promising in regards to the characterization of the grabbing motion in the context of a wristband. After a certain number of templates of the same activity is generated, we apply dynamic time warping [12] to identify possible matches. We assume that the same motions produce similar outputs which only differ in respect of their length due to the varying speed the activity was performed. Thus we choose dynamic time warping as it allows us to match time series of different lengths. Dynamic time warping works by finding a path between two time series that have the smallest distance. The minimal distance is found by first initializing the distance from every point in series A to the first point in series B to $\mathtt{infinity}$ and vice versa. Afterward, the algorithm iterates over the combination of all points in both series and calculates their distance by using a $\mathtt{cost}$ function (in our case Euclidean distance). The function compares single points and the cost of the path leading to the previous points (recursive):

$$d(i,j) = \mathtt{cost}(i,j) + \mathtt{min}(d(i,j-1), d(i-1,j), d(i-1,j-1))$$

By considering three preceding options that lead to i,j, the algorithm can cope with different lengths of series. The extracted templates slide over an unlabeled dataset to detect the time when an activity occurs. In this context, we try to find the position of the template with the smallest deviation while assuming that at least one activity occurs in the unseen data. This is our base method which will be adapted in the following methods and made more complex.

3.4.2 *Activities of daily living dataset*

For our activity of daily living dataset, we focus on a small subset of activities where the test subject is lying. By only considering the human body pose of lying, we expand the variation in our experiments thus looking at a wider range of use cases. For the alignment of the data we once again use zerolines in the acceleration data. We consider the moment the participant starts their first activity as we do not have long walking distances which can be used to align peaks. After alignment, the plot of the data with markers for labels is consistent and shows labels at the correct point in time. In contrast to the picking scenario, the dataset of our ADL scenario has multiple activities per run. Therefore, we cannot pick the best match but rather try to find a set of best matches, which changes the specific implementation of dynamic time warping we need to use. Due to the similarity of the activities, we were not trying to match the activities themselves but rather the sub-activity of raising an arm and reaching towards a glass of water, a plate, or a pill bottle. This motion is similar to the picking motion but contains more variation as the environment is more dynamic. To get the full activity, we would need to consider the

visual aspects of the data as well. Afterward, in the annotation phase, a person labeling the dataset can manually distinguish which activity to assign to the template matching results. By applying this trade-off, we minimize the amount of pre-labeling for the user as we do not have to have templates for each activity. Methods used in the picking dataset were extended and new ways of transforming the data as well as evaluating were added. Figure 10 shows the configurations which we tested with our new methods. They can broadly be classified into parameters that influence the input data format for our algorithms and different settings for selecting candidates for our final results.

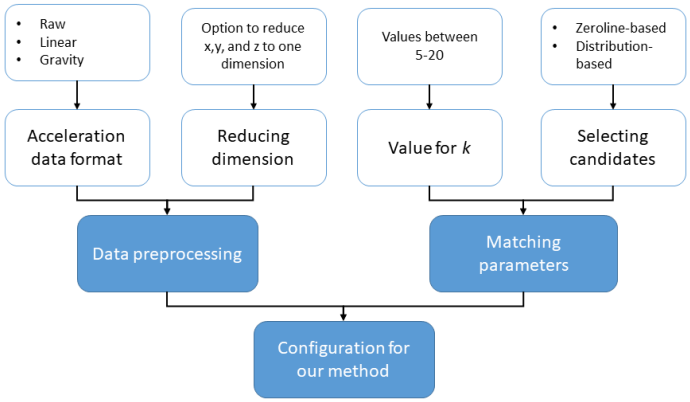


Figure 10: ADL experiment settings. Each parameter is set for a specific configuration of the matching algorithm.

For the pre-processing of the acceleration data, we consider two parameters that we can alter to change the representation of the data. One parameter is the type of acceleration is used (linear, gravity, or raw data), the other parameter is an option to reduce the three axes of acceleration data into one.

ACCELERATION DATA TYPE We distinguish between two ways of transforming the acceleration data we collected. One option is to either use the **linear** acceleration or **gravity** of the data by applying a low-pass filter. We also test the **raw** data collected by our Android application to match the activity.

REDUCING DIMENSIONS This option specifies whether the acceleration in the x, y, and z dimensions should be reduced to a single value. In preliminary tests for the data, we could often see that matching templates with only one dimension of data yields better results. One reason for these improved results may be the loss of orientation information when reducing dimensions which may yield a more generic model. We reduced the dimensions by interpreting the x, y, z acceleration values as a vector and calculating its vector length.

Once the data has been transformed, we apply dynamic time warping and then select top_k candidates that we consider as our possible labels. Therefore, our method also has two configurations for the candidate selection: the k-value in our top_k selection and the method of selecting the candidates.

VALUES FOR K As we are considering a sequence of activities per recording, we retrieve the top_k best matches. For the evaluation, we tested values for k ranging from 5 to 20.

SELECTING CANDIDATES We apply two different methods for picking the best matches as choosing only the points in time with the lowest distance does not yield the best results. Each method is described in more detail below.

After deciding on the parameters, we match the data with the dynamic time warping algorithm. In this case, we use the subsequence matching variant of dynamic time warping as described in [67]. To get candidates for our labels, we first match the template against the data. Then we explore two variants of picking potential matches for our activity. In the first method, we additionally match a zeroline template against the data. This is used for the sequences of data that do not contain any of the activities we try to find. We assume that the distance of a zeroline template match to these sequences is smaller than the distance of the templates for our target activities. This way we reduce the number of candidates we have to consider when matching our correct templates, as we now have a notion about the potentially non-relevant parts of the data. For all matching candidates with the activity templates, we take the top_k matches with the smallest distance and return them to the program. In our second method of finding candidates, we use another simple assumption. When considering candidates, we look at the current selection we made. A candidate for a match is only added to the list of top_k candidates if it is not within a distance δ of any currently selected top_k candidate. We set δ to two seconds as our activities are not performed within a shorter period of time. Both methods for selecting candidates can be seen in Algorithm 1 and Algorithm 2.

```
Data: InputData, Template, Zeroline, Options, K
Result: Distances
template_distances = dtw_subsequence(InputData,
 Template);
zeroline_distances = dtw_subsequence(InputData, Zeroline);
candidates = List();
foreach dist_template, dist_zeroline in template_distances,
 zeroline_distances do
    if dist_template < dist_zeroline then
        candidates.append(dist_template);
    end
end
return top_k(candidates, K);
```

Algorithm 1: Method based on zeroline for selecting candidates

```
Data: InputData, Template, Options, K
Result: Distances
template_distances = dtw_subsequence(InputData,
 Template);
candidates = List();
foreach dist_template in template_distances do
    if dist_template not within range of candidates then
        candidates.append(dist_template);
    end
end
return top_k(candidates, K);
```

Algorithm 2: Method based on distance for selecting candidates

Once the candidates are determined, we evaluate them based on the distance to the actual labels.

3.4.3 *Kitchen dataset*

Focusing on the *CMU-MMAC* dataset, due to its complexity, we have to consider different steps. In contrast to the picking and our ADL example, people switch between the left and right hand in this dataset, which means that it is also necessary to identify which hand was used for the current activity. In our ADL dataset we also had the case of using both hands for activities, but we could mitigate that uncertainty by additionally annotating the hand which performed the activity and then matching the activities separately. We do not have this information in the annotations of the kitchen dataset. This dataset already provides aligned data making the process easier for us. Therefore, we unify the data of the same sensor type of both hands so that the current activity is represented by a single vector. Considering the corresponding labels, it stands out that the described activities cover several motions, e.g., grabbing is only a sub-activity. Therefore, still focusing on acceleration data and considering the corresponding ground truth to extract the templates, we segment the

data of a template into small windows to compute features that have a stronger expression concerning more complex activities. This includes the used energy (Fourier transform) and median absolute deviation. Due to these high-level labels, several different activities may cover common sub-activities, e.g. taking a pot or turning on the stove includes grabbing. Therefore, we also investigate if the extracted templates have a label independent correlation. We assume that the extracted templates could be grouped into activities that are specific in their motion and not in their semantic. For that purpose, we apply agglomerative clustering to group the templates where the distances between the clusters are the result of the dynamic time warping. Detecting the motion similarity between certain activities may not only enable us to generalize activity labels. It also facilitates the construction of more robust templates due to the varying executions which in turn helps us to avoid overfitting.

For the experiments, we perform leave-one-out cross-validation. Thus, we extract templates from n-1 datasets, and apply them on the remaining one.

3.5 EXPERIMENTS

In this section, we focus on the performance of our labeling support tool to see if it is a feasible approach to be used on a greater scale. This involves an evaluation against ground truth data to establish how well the tool is able to find labels. Since we established different methods for our datasets, we also evaluate them differently. We evaluate our approaches mainly in regards to how close our estimated label is located relative to the correct label. A value that describes how close the estimate is towards the correct label (in our case called delta) is more in line with the task. To give readers an intuition of the results, we still added measures for recall for appropriate experiments.

3.5.1 *Picking dataset*

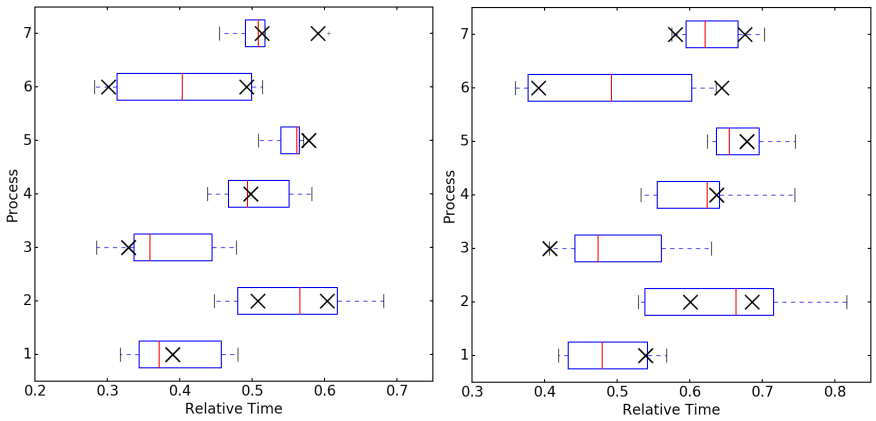
In our first experiments, we only focused on the grabbing activity in context of the acceleration sensor data that correspond to the wristband. Thus, we want to investigate the feasibility to apply template matching across different people to identify certain activities, where in turn, the result should be used to provide recommendations concerning the labeling of the video recordings.

For that purpose, we first apply our introduced method on our picking dataset. We extract the grabbing motion templates from all except one dataset, with each set covering one complete picking process (which may contain a double pick). Then we measure the temporal overlap of the estimated and the actual grabbing motions. For the average overlap per dataset, we take the best match (i.e. the match with

Dataset	1	2	3	4	5	6	7
Overlap	0.43	0.67	0.78	0.52	0.72	0.74	0.99
Motion [s]	5.02	2.49 2.22	2.55	4.23	2.86	2.43 4.11	2.04 2.60
Δ Start [s]	1.41	1.89 1.81	0.91	0.86	0.71	2.88 2.61	0.65 2.91
Δ Duration [s]	1.65	0.74 0.68	1.40	1.46	0.63	0.68 1.52	1.99 1.43

Table 1: Recognition performance of template matching for picking dataset. The overlap (avg. 69%) is excluding outliers and represents only the best match within a dataset. Cases 2, 6, and 7 contain two grabbing activities.

the least distance) for each template. Afterward, we select the most promising subsets of matches and use them to calculate the average overlap for each test dataset. The most promising subset of matches is determined by evaluating all the subsets of the matching results with k elements and then selecting the one with the greatest overlap among itself. Empirical results show that a value of k=6 yields the best results on our dataset. Table 1 summarizes the results and reflects that we are able to detect nearly all grabbing motions, but have an issue concerning the accuracy of the start and stop boundaries. Indeed, inspecting the individual results shows that our assumption, that the searched activity has to have the same length as the considered template, leads to inaccuracy.



(a) Result Start (b) Result End

Figure 11: Overall estimate of grabbing start (a) and end (b) point for picking dataset. Cases 2, 6, and 7 contain two activities and therefore also two crosses in the plot.

Figure 11 describes in detail the recognition and distribution results for all start and stop times. We want to emphasize that the x-axis does not represent the recognition rate but the relative duration of the whole process. Hence, the box plot represents the time inter-

Acceleration	Reduce dim.	Approach	Subject 1	Subject 2
Raw	Yes	Zeroline	[0s - 28.9s] $\tilde{x} = 14.7s$	[0s - 39.3s] $\tilde{x} = 7.9s$
		Delta	[0.2s - 5.1s] $\tilde{x} = 0.6s$	[0s - 1.4s] $\tilde{x} = 0.5s$
	No	Zeroline	$\emptyset$	$\emptyset$
		Delta	[0.1s - 14.5s] $\tilde{x} = 2.2s$	[0s - 39.3s] $\tilde{x} = 7.7s$
Gravity	Yes	Zeroline	[0s - 20.1s] $\tilde{x} = 10.2s$	[0s - 39.3s] $\tilde{x} = 11.1s$
		Delta	[0s - 1.5s] $\tilde{x} = 0.27s$	[0s - 12.5s] $\tilde{x} = 0.8s$
	No	Zeroline	$\emptyset$	$\emptyset$
		Delta	[0.2s - 10.9s] $\tilde{x} = 1.6s$	[0.2s - 41.3s] $\tilde{x} = 3s$
Linear	Yes	Zeroline	[1.2s - 82s] $\tilde{x} = 24.2s$	[0s - 48.4s] $\tilde{x} = 8.2s$
		Delta	**[0.1s - 1.5s]** $\tilde{x} = 0.8s$	**[0s - 1.4s]** $\tilde{x} = 0.5s$
	No	Zeroline	$\emptyset$	$\emptyset$
		Delta	[0.4s - 9.6s] $\tilde{x} = 2.2s$	[0s - 21s] $\tilde{x} = 3s$

Table 2: Results for matching activities of daily living. For each case, we report min and max distance to activities and median distance. The bold values show the best results.

val where we assume the start point and respectively the stop point, for the activity that should be recognized. Every box represents the best match for the templates where the x markers show the actual point in time of the grabbing motion. As there can be two grabbing motions in a dataset we plotted both positions. The boxes provide an interesting insight concerning the reliability, i.e., most of the extracted templates were able to identify the correct area of a certain activity across different recordings of the same process.

3.5.2 *Activities of daily living dataset*

For our activities of daily living scenario, we evaluate different settings of the algorithm. We first consider the top_k best matching results with k = 10. After finding the best pre-processing settings, we further test different values for k. We compare different methods of pre-processing the data as well as ways of choosing the best matching candidates. Data can either be used unchanged or be transformed to get linear acceleration or gravity information. For each of these types, we also compare the performance of using all three axes of the data as well as reducing it to one dimension. Finally, we pick best matches by considering one of two options. One option looks at all the matches that have a smaller matching distance value than the zeroline template at the same point in time. Of those matches, the k smallest values are chosen. The other option is picking lowest matching distance values that do not lie within two seconds of each other (see Section 3.4 and Algorithms 1 and 2). All distance values are calculated using the matching results with ten different templates and then summing up the distances. The results can be seen in Table 2. We can

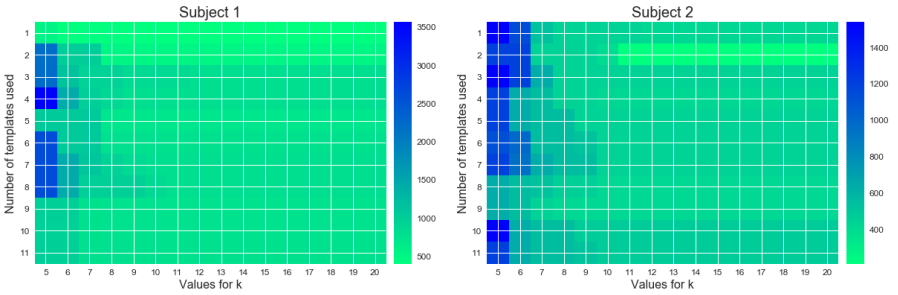


Figure 12: Results of matching activities of daily living with different numbers of templates used for matching and different values for *k*. The color shows the average distance (in ms) of a match to a label.

see that the best performing combination of settings is the usage of linear acceleration with the dimensions reduced to one and using a delta-based top_k approach. This yields a median error of 0.8s and 0.5s for both subjects. One additional finding is that the approach of using the zeroline for choosing possible matches is not performing as well as the delta-rules-based approach. It does not even return values for some of the use cases, namely the cases that do not reduce the values to one dimension. A possible explanation is that a zeroline returns a smaller distance over all the datasets. Further analysis of the approach will be done just on the best performing setting, which is using linear data with the dimensions reduced to one and using a delta-rules-based candidate selection.

Figure 12 shows the results of different evaluation parameters using the previously mentioned best-performing methods of data transformation and candidate selection. It can be seen that the amount of templates used to find matches is not affecting the results significantly. Instead, the chosen k is more important to get a reasonable result. It can be seen that just using two templates and then picking the top_{10} matches is sufficient to find activities within a reasonable margin of error.

To further evaluate the performance of our method and provide another form of intuition, we also show recall and ROC curves for the results. The performance is split up for each hand separately in this scenario, to show the differences in the results.

The plot in Figure 13 shows how the recall for each separate hand is changing with different values for k. It can be seen that recall for the left-hand is behaving differently than the right-hand evaluation. The ROC curve in Figure 14 also reflects this fact. This is most likely due to the fact that the primary hand of the subjects is the right hand. Therefore, the motion of the left hand is not as consistent as the motion of the primary hand. Overall we can conclude that with appropriate values for k, the method yields acceptable results for labeling purposes.

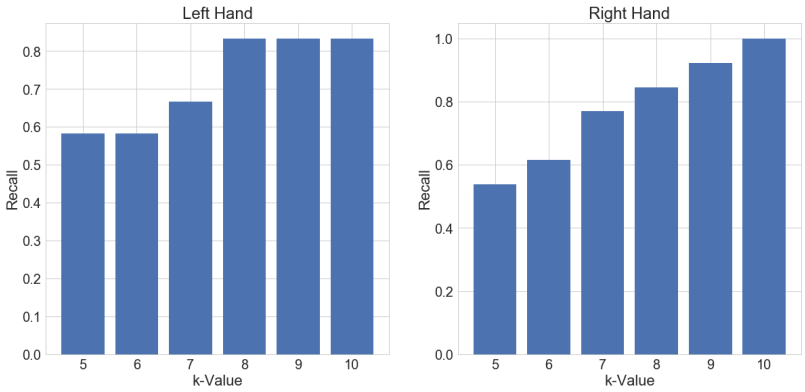


Figure 13: Recall of the results of both hands depending on the value of k that was used for candidate selection. Overlap with the ground truth labels is counted as a True Positive.

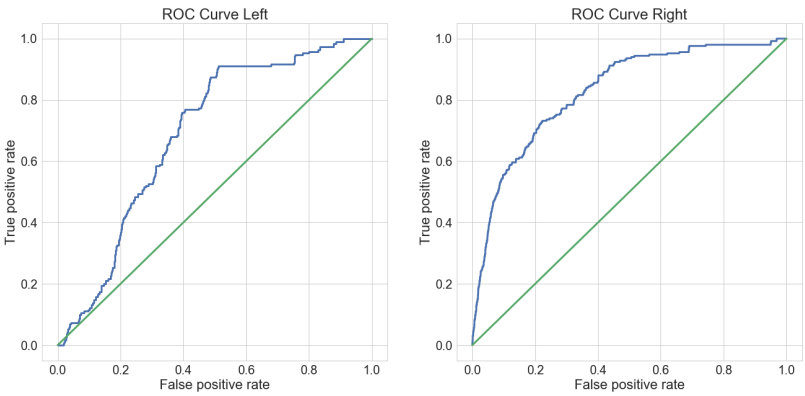


Figure 14: ROC curve for both hands. Without the candidate selection, this plot shows the overall performance of the Dynamic Time Warping algorithm. Again it can be seen that the performance for the left-hand data is not as consistent as the performance of the right-hand data.

3.5.3 *Kitchen dataset*

Considering the *CMU-MMAC* dataset, our first results were insufficient because different activities covered similar arm movements. For instance, the extracted templates of the activity *take oil* are also wrongly recognized as *put oil into cupboard*. Thus, we tried to cluster the activities based on their similarity to get an insight regarding their meaning. Figure 15 illustrates the clustering results of one sample set. It is striking that some activities that use items within a similar location are ending up in the same cluster fairly consistently. For instance, we can observe that motions like taking the big and small measuring cup are very similar. In contrast, the fork and the scissors for instance are both located in a drawer but end up in the same cluster fairly late. We believe that this is most likely due to the fact that the activities are more variable in length than they are in our own datasets. Even though dynamic time warping is able to handle different lengths of time series, it is still very likely that the distances of short templates are generally smaller and thus end up faster in clusters than the longer activities. This is for instance the case for taking the baking pan from the oven.

Figure 15: Dendrogram of the clustering of the templates in the kitchen dataset. Marked boxes are activities using the same item.

3.6 CONCLUSION AND FUTURE WORK

In this chapter, we investigated the first step of a typical HAR pipeline: annotation of data. We explored the possibility of a smart data annotation tool that provides labeling recommendations based on the already labeled acceleration sensor data. These recommendations can be used to speed up the annotation of video and acceleration data by finding possible activities in the dataset and showing these guesses to the user. She or he then only has to look at the recommendations and does not have to look through the whole dataset to find the activ-

ities. For that purpose, we performed experiments to investigate the feasibility of applying template matching in context of dynamic time warping to recognize certain activities across different processes and people. In this context, we focused on the acceleration data of a wristband and smartwatches to recognize certain activities. It has emerged that it depends on the granularity of the considered activity labels which recognition technique is promising. Hence, activities that actually consist of several sub-activities may have to be labeled separately at the beginning. To further investigate this, we adapted the experiments to another dataset, containing activities of daily living. Here, we looked at activities that involve eating, water consumption, and medicine intake. We can show that these activities, which all involve similar movements, could be recognize fairly consistently by applying template matching and only taking into account the initial arm movement part of the activity. In this context, we also showed that clustering existing templates from a labeled dataset allows inferring similarities in motion from semantically different activities. This can be considered as a starting point to construct more robust templates. The clustering results also yield more information for a specific motion, which in turn reduces the need to perform a certain activity more frequently to get enough characterizing information. In contrast to other approaches [45, 84], we need significantly less data to guess the correct time frame of a specific activity. Another aspect we looked at in this work is the ability of acceleration data to help annotating video information. Here we could see that acceleration data on its own is not capable to find all types of activities. Especially for the ADL dataset we could see that scenarios exist where movements are too similar to fully distinguish different types of activities.

In our future work, we want to focus on the problems which came up during our investigations. This includes the recognition quality of the boundaries of activities due to the limitation of a predefined template length. In addition, the fact that we considered only acceleration data so far is another possible source for inaccurate results. Thus, considering further sensors may also increase the recognition accuracy. For that purpose, we want to enhance our own dataset concerning the number of instances but also regarding the considered activities since it turned out that the considered activity level is essential. Another important step in future work is a user study with a group of annotators that measures annotation time as well as the agreement for a set of labeling tasks with and without the recommendations of our proposed solution. Results from such a study could also point us to other improvements of our method that we have not considered yet.

4

SEGMENTATION OF SINGLE ACTIVITIES

In the previous chapter, we have looked at methods that enable a faster annotation process for activity recognition pipelines. As we have seen, activities usually vary in the amount of time they span. This poses researchers with the challenge of finding segments in one recording of activities.

4.1 INTRODUCTION

In the field of activity recognition, multiple methods for recognizing the sequence of activities can be applied. One could try to transform the sequence of raw data into equal units, e.g. with a sliding window approach, first. Then, each window is classified independently, giving the end result. Another approach could try to first segment the data into separate sequences which do not need to have the same size. It can be seen that especially scenarios which involve activities with a huge variety can lead to a big range of durations. This chapter deals with exploring methods for automatically segmenting data in the context of activity recognition.

We can first try to establish the different challenges that impact the segmentation:

INTERLEAVED ACTIVITIES

In many scenarios activities are performed in short succession. They overlap, making it very difficult to distinguish them from one another, even for human annotators. We can assume an example from the CMU-MMAC dataset that deals with food preparation:
Right after mixing brownie batter, a test subject puts away the spoon they used. The single activities they performed were the *mixing* and the *put_away* activity. However, the motion of mixing batter and then putting away the spoon can be seamless. This makes it difficult to establish the point in time when one activity ends and another starts.

SEMANTICALLY AMBIGUOUS ACTIVITIES

Often, datasets can be made up of annotations that are not atomic. Specifically each label can share meaning with other labels in the dataset. Labels can be defined with specific rules in mind which theoretically allows for models to predict a wider range of activities than initially learned. In the case of the CMU-MMAC dataset, activities are made up of a verb, a subject, and, optionally, a preposition and an object (e.g. *put_cup_on_table*). It can be sensible to learn single parts of the activities independently. However, we run into one issue when we consider

activities that use the same verb but involve different motions or even slightly different meaning. An example would be the motions for *open_fridge* and *open_browniemix* as both denote the opening of the corresponding subject but with a very different meaning regarding the actual motion. The activities *open_fridge* and *close_fridge* however, use different verbs but have a very similar motion. In segmentation, this can cause issues when we try to utilize the similarity in motion to find the start and the end of an activity. Here, we may not be able to rely on the label name but rather on domain knowledge.

SENSOR PLACEMENT

Different publications have already explored the importance of sensor location for detecting specific activities [87]. In the case of the CMU-MMAC dataset, we can choose from a variety of sensor positions as the dataset provides inertial data from up to nine different positions (including torso, forearms, and shins). However, often researchers may want to use a minimal amount of sensors for their models (e.g. when a system is supposed to run on consumer hardware where only one sensor is provided). This causes issues for some activities that can be difficult to detect with wrong sensor placement. Short walking sequences, for instance, can be detected more easily with sensors on the legs while wrist-worn sensors may miss the short motions entirely or disregard it as noise. When trying to pick only one sensor placement, we have to consider the most promising candidates for predicting the biggest set of activities.

This work explores different methods for segmentation, specifically for the CMU-MMAC dataset. We want to evaluate if common methods for segmentation are applicable for this fairly difficult dataset. Specifically, we look at inertial data-based segmentation as we do believe that changes in movement are an important factor for detecting activity change.

This work has not been published in previous publications.

4.2 RELATED WORK

Segmentation of time series data is a topic that can be applied in multiple fields. Human activity recognition specifically needs methods for segmentation in order to reconstruct a series of activities that occurred. To get an overview, we look at specific methods to segment activities, specifically using inertial data as this is one of our main modalities within the datasets we consider. Many methods are evaluated on segmenting gait cycles. Since single steps are often used for traditional human activity recognition, gait cycles are a useful unit for segmentation and therefore can be seen as a significant part of the pipeline.

Agostini et al. [2] show one such approach, where foot switch signals have been used to identify different parts in a walking cycle. By

measuring contacts made with the floor of different parts of a human foot they are able to properly identify parts of the cycle. This shows, that especially cyclic activities may contain distinct features that can be utilized for segmentation of activities. It still needs to be seen how this relates to more complex activities. The researchers went on [3] to extend their research by switching to magneto-inertial sensors and analyzing different test subjects.

Some researchers [16, 38] consider energy spikes as a good indicator for change in activity. Broadly speaking the assumption is, that change in activities is associated with a certain amount of movement that a person has to perform in order to change activities. When the energy from raw inertial data is calculated, it enables the researcher to quantify the level of movement. This approach works in the presented work where the authors consider simple locomototive activities. In our work, we want to test this approach on the more complex CMU-MMAC scenario while also building on top of it.

Other work [81] uses Hidden Markov Models to properly segment activities. A focus was put on locomotive activities like walking, standing, lying, and climbing stairs. For that purpose the system is able to perform decently, thus we test a simple HMM approach as well.

Dynamic time warping, as seen in the previous chapters, was also proposed for sequence segmentation [9]. Here, the specific task was geared towards segmenting strides of human walking cycles. By annotating and extracting templates from a controlled data gathering setup (i.e. walking in a straight line for a given distance), multidimensional dynamic time warping can be applied to recognize strides in human movement. This work also shows the influence of different types of inertial sensors (namely accelerometer and gyroscope) as it compares the results of each sensor on its own with the combination of both sensors.

4.3 METHODS

For this problem, we consider multiple approaches that may be applicable to our scenarios. In this subsection, we elaborate on the broad ideas for each approach we test, and details for each method. These methods will later be evaluated in the Experiments section. Figure 16 shows a sample plot of energy values (using the three axes of gyration and acceleration respectively) and the actual starting and ending points of activities. It can be seen, that some spikes co-occur with the borders of activities. Therefore, we try to use energy values for our segmentation methods.

4.3.1 *Hidden Markov Models*

Typically Hidden Markov Models (*HMMs*) are used to model the change of hidden states by examining a dependent observable variable. To illustrate how HMMs work, we can look at one typical use

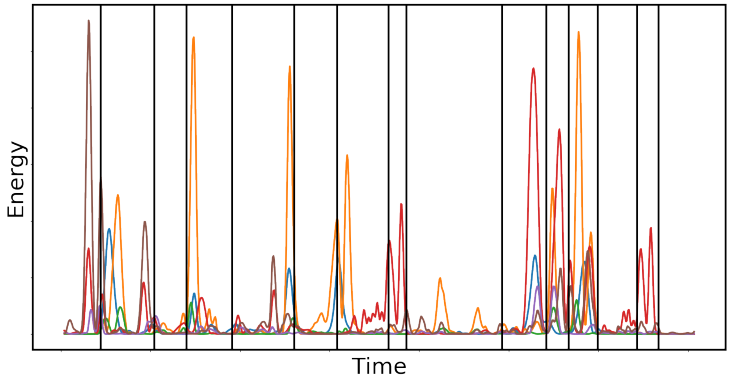


Figure 16: Illustration of energy values and corresponding segments

case: speech recognition. Here, programs receive audio waves as an input and try to estimate the underlying syllables that lead to these sounds. Hidden Markov Models may model the syllables of a phrase as the hidden state and the sounds that are emitted as the dependent and observable variable. In this way, the syllable that the algorithm wants to estimate depends on the previous hidden state and the current observation.

We try to map this approach to our problem. Here, we look at this challenge in two ways: In both cases, we consider the inertial data as our observable variable. Then, we can either decide to model each type of activity as a separate hidden state or we model the sequence with two states: *in_activity* and *change_activity*. The difference between the approaches is the amount of information that we assume to be extractable from the underlying data.

4.3.2 *Machine Learning Approaches*

Another fairly intuitive method would utilize machine learning for the segmentation task. In that case, we assume that we can find a feature representation that allows us to distinguish the classes *in_activity* and *change_activity*. For the representation, we choose a sliding window approach where we calculate sets of features from the raw data. Then we use a typical learning algorithm like a Random Forest to classify the windows. This approach allows us to leverage our previous expertise in feature engineering for activity recognition.

4.4 EXPERIMENTS

We test all our approaches on the CMU-MMAC dataset. To evaluate the methods we calculate recall, precision and F_1-score. We are only interested in the inertial information to segment our data. Based on the findings of [38], we want to use energy values for our methods as well. To calculate the energy values, we create sliding windows from the inertial data. The window size is set to 200ms and con-

secutive windows have an overlap of 90%. This way we have an effective sampling rate of 50Hz. To create the new annotations that distinguish between *in_activity* and *change_activity*, we looked at all ending-timestamps of the given activity labels. Since the activities in the CMU dataset are consecutive and do not contain holes, we have to manually assign a length to the change class. We chose 200ms, which is the window size we consider for the energy calculations.

Initially, we analyze how energy values can be used with a simple Hidden Markov Model approach. In Figure 16 we can see some regularity of the segment change and the energy values. Based on this, we apply the HMM algorithm as a first experiment. We consider the subset of all Brownie recipes and test two approaches: one of them uses two states where we differentiate between *in_activity* and *change_activity*. For the other approach, we only look at the verb part of the label and use theses as the states for our HMM.

In the training phase, we take a subset of all scenarios (in this case 25 of the Brownie scenarios). Out of these, we train on the first 24 scenarios and evaluate on the last one.

Type	**Precision**	**Recall**	**F_1-score**
in_activity	0.94	0.87	0.90
change_activity	0.06	0.14	0.09

Table 3: Confusion matrix for the Hidden Markov Model

The results as seen in Table 3 are not satisfactory. We then take a closer look at the results by examining a sequence diagram of the actual state compared to the prediction of the model.

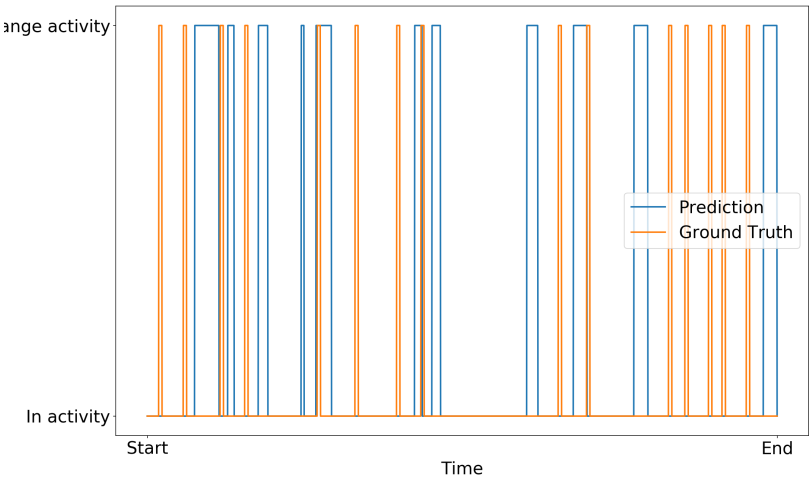


Figure 17: State sequence for a subset of the whole test scenario.

Figure 17 shows the predicted and the actual states for a subset of the whole test scenario. We can see some interesting findings in the predictions. The model sometimes predicts longer sequences of changing states. This is of course not intended in our scenario and

Type	Precision	Recall	F_1-score
in_activity	0.98	0.62	0.76
change_activity	0.02	0.39	0.04

Table 4: Confusion matrix for the Hidden Markov Model using verb labels

could be fixed with post-processing e.g. by choosing the center or the start of each sequence as the changing state time. Another finding is that the model overall predicts fewer changes than the actual amount present in the data. If we count the total number of predicted changes and compare it with the ground truth we get a ratio of 1 to roughly 2.2. This means, our model predicts a change more than twice as often as it is the case. However, we can also see that there are instances where the prediction overlaps fairly precisely with the actual change. On the same note, the long sequence without a change roughly in the center of Figure 17 is mirrored in the prediction of our model.

For our second approach with HMMs, we adapt the scenario by replacing the states we proposed initially with the verb-part of the annotation. Thus, the model now has ten different states it can predict. To evaluate the results, we transform the predicted sequence to the format used beforehand, as we are only interested in the change of state.

Table 4 shows the results of our second HMM approach. Here, we can see that the numbers drop compared to our first HMM approach. Apparently, using the verb part of the label does not improve our overall prediction but rather worsens it.

Finally, we examine the machine learning approach. We test out a few classifiers like a Random Forest, a Support Vector Machine and others. For this approach, we evaluate different types of features. Like before, we used only energy values in the first run. At a later stage, we instead use sliding window features to see if the results improve.

For the training, we first split the data into train and test sets with a ratio of 80 to 20. Since the change class is very rare, we up-sample these examples in the training set. This should prevent our algorithm from always predicting the majority class.

Table 5 shows the results for the different settings we tested. It is important to note, that the numbers shown are the weighted averages. Even though the results look good at first glance, the precision for the change class is very low. Replacing the energy values with the window features does not yield much of a difference. Interestingly, the Random Forest benefits from the energy values, while the other algorithms return very similar numbers. Throughout all the algorithms it is not possible to determine a good segmentation of the data just by applying machine learning.

There may be multiple reasons for the poor results we get. One main reason is that the activities are performed in a very fluid way, making it harder to detect when one activity starts and another one

Algorithm	Precision	Recall	F_1-score
Random Forest (energy)	0.96	0.72	0.81
Random Forest (raw)	0.95	0.64	0.75
Naive Bayes (energy)	0.94	0.84	0.89
Naive Bayes (raw)	0.94	0.71	0.80
SVM (energy)	0.94	0.69	0.79
SVM (raw)	0.94	0.67	0.77
Decision Tree (energy)	0.95	0.55	0.68
Decision Tree (raw)	0.95	0.58	0.71

Table 5: Performance for different algorithms and settings, as a weighted average. This table is a good example, why weighted numbers are not necessarily the best representation for results, as they can mask problems.

Algorithm	Precision	Recall	F_1-score
Random Forest (energy)	0.08	0.81	0.15
Random Forest (raw)	0.05	0.65	0.10
Naive Bayes (energy)	0.04	0.18	0.06
Naive Bayes (raw)	0.04	0.37	0.07
SVM (energy)	0.04	0.39	0.07
SVM	0.04	0.42	0.07
Decision Tree (energy)	0.04	0.56	0.07
Decision Tree (raw)	0.05	0.62	0.08

Table 6: Performance for different algorithms and settings for the change class. Here, the actual issues of the segmentation method is visible.

ends. Another explanation could be that some activities involve movement that is not captured via the wrist-worn sensors, e.g. walking. Thus, among the given sensors, no particular change is visible.

4.5 CONCLUSION

We could see that segmentation in activity recognition is a non-trivial task, specifically when looking at complex datasets like the CMU-MMAC dataset. Our results show, that traditional methods for segmentation may not work that well in complex settings.

However, it can be worthwhile to consider different approaches in future experiments. While out of scope in this work, the addition of visual features for the segmentation may aid our task. Examples for visual features could be the appearance and disappearance of objects in a frame or the interaction of the test subject with objects in their environment. Here however, it would make sense to first analyze the activity change in a qualitative way. This way we can determine how well video as a modality suites the task, e.g. if specific objects

are a good indicator for activity change. Also, for object-based segmentation, overfitting to the recording environment may be a bigger challenge compared to the inertial data.

Our initial core idea for segmentation of the data is to aid the classification of the activities. But we could also redefine the order of operations within the pipeline to achieve our goal by leaving out the segmentation. Let us assume that we divide the data in sufficiently small windows. If we first generate features from our dataset that allow us to classify activities reliably and independently of the length of the activity, segmentation could be replaced with a post-processing step. In practice, we would build sufficiently sized windows from our dataset (ideally not longer in duration than our shortest activity). Afterwards, we can classify each window independently, assuming we do not need the previous windows' information. From this series of predictions, we could create the most likely sequence of activities for each single run. Using smoothing and other similar methods, we may utilize the predictions to additionally estimate start and end of an activity. For this approach to work, we have to evaluate if classifying windows in itself is feasible with a certain margin of error.

5

MULTIMODAL PREDICTION IN AN INDUSTRY SETTING

For this work, we take a look at activity recognition in a setting typical for industry use. We emulate a warehouse picking scenario and try to detect picking motions as this is a typical use case for warehouse management systems.

5.1 INTRODUCTION

In the field of modern warehouses, a lot of attention is put on improving the process of order picking regarding accuracy and time in order to save on costs[23][42][89]. Order picking denotes the process of fetching items that are contained in a customer's order for retail or parts that are needed for further assembly in a factory. Since picking items is often one of the first parts in a longer business process, errors in this stage may only be detected in a later stage, leading to high costs for a fairly minor error. Modern wearable technologies like smartglasses, smartbands, and smartwatches can aid the picker in their task, helping to reduce errors made in the process. We can see two main ideas here: the wearable device can on the one side be retroactive by notifying the worker if they picked the wrong item. Additionally, it can also work proactively by helping the picker to find the correct shelf in a potentially huge warehouse, therefore reducing the time a worker has to spend localizing the item (though skilled workers usually are so experienced that the system works more like a sanity check). On the other side, wearable devices can free up the workers' hands when compared to traditional scanners. This is especially useful for training new employees who have yet to learn every single step in the picking process. We can identify two types of approaches to improve the picking process in a warehouse scenario: One type aims to equip the pickers with tools to speed up their workload. In an ideal setting, it could even remove some of the work, e.g. by automatically scanning the picked item as it is lifted from the shelf. This could be done by equipping pickers with voice control systems[63] or by giving the worker wearable devices that directly scan the item [96]. On the other end of the spectrum, the warehouse itself could be augmented to help with the localization and accuracy of picked items. An example for this would be the highlighting of shelves that contain the correct item and additionally create projections of the needed amount [37]. Similarly, depth-sensing cameras could be added to shelves to detect picking for each specific item location [55]. This work is following the approach of the former category. We base our choice on the assumption that a solution which equips the worker can easily be adapted to other warehouses without the need for possibly costly hardware installations on-site.

This work explores a wearable-aided picking detection that could potentially be used in a warehouse environment. Specifically, we consider smartglasses in combination with a smartwatch as our main source of data. From these devices, we extract inertial data (i.e. mainly acceleration in addition to gyration and magnetic field) as well as first-person video information. With both modalities present at the same time, we try to handle each shortcoming. For video data, we have to consider the fact that a lot of the relevant action is not captured, which is a problem that is especially emphasized by the limited field of view of the camera built into most smartglasses. In the case of inertial data, we may have to deal with the issue that simple arm movements (e.g. while walking) can be classified as a picking motion. Additionally, we are mostly interested in finding the correct start of the picking motion since this would give a potential warehouse system the most time and information to do further checks regarding the correctness of the item. For these purposes, we pose two research questions:

RQ1: Can the combination of inertial and video data be used to classify grabbing actions?

RQ2: If so, what subset of features is best suitable for that task?

To answer these questions, we created a dataset for a typical picking scenario. It includes multiple participants performing different picking tasks in a simulated warehouse environment. Afterwards, we analyze whether we can learn to distinguish grabbing from non-grabbing actions within this dataset. In this case, the simple act of picking an item would be more in line with an action instead of an activity hence we use the term action in this part of the work.

We structure this analysis as follows: In Section 5.2, we describe existing work in the field of multi-sensor fusion and feature selection in the context of activity and action recognition. Afterwards, we describe our dataset in Section 5.3. Section 5.4 covers our methodology with a focus on the features we select for our experiments, which are described in Section 5.5. Then we conclude the results in Section 5.6 and give an outline for future work.

This chapter reflects the content of our previous publication [33].

5.2 RELATED WORK

Modern warehouses often rely on RFID and/or QR codes to validate orders[23]. While these approaches are very precise, the validation happens at a late stage. By using wearables, we aim to register the picking action earlier. This allows us to notify the picker about a correct item position or even recognize a pick from a wrong shelf before it fully occurred. In this work, we are dealing with action recognition on multi-sensor data and the influence of different extracted features

for recognizing the action. For that reason, we focus on related work of activity recognition but also multimodality as feature selection, extraction, and fusion are commonly covered by these fields.

Researchers already showed that using acceleration data with a sliding window approach works for human activity recognition[51][74][81]. Typically, this targets the recognition of activities like walking, jogging and climbing stairs. A special focus is put on the position of the sensors and how it influences the classification results[74]. Sensors are often placed on the legs, the arms, and the torso of subjects and then evaluated either separately or in combination. Commonly, the considered features correspond to the time and frequency domain and have been shown to work for these activities. These features are then classified with machine learning algorithms like Decision Trees[51], Hidden Markov Models[81], and kNN[74]. Recently, Ordóñez et al.[71] also used Neural Networks for human activity recognition. Indeed, they are able to show that by adding a new modality (e.g. adding gyration data to acceleration data) to a network, new features can be extracted from it without any need for pre-processing. Many of the features considered in previous work are extracted from a long timespan (typically 1-3 seconds). As we are considering actions instead of activities, which span a much shorter time, we keep the window size small and chose a bigger overlap of consecutive windows. Therefore, we apply similar approaches to smaller data to test if these experiments still hold for our grabbing scenario.

Analyzing inertial data for activity recognition only covers half of our analysis. We also want to consider the video sensor for our classification experiments. Combining different kinds of sensors to create a multimodal dataset has been the focus of various previous studies[88][19][83]. These focus on activities like cooking[88], sport activities[19], and office work[83]. Using these datasets, researchers fused both modalities to recognize activities[84]. Problems that arise in this context are combining sensors with different sampling rates to create one time sequence. Solutions for this problem include downsampling inertial sensor data to fit the frame rate of the video[84]. This approach does not work with features extracted from windows which we use for our inertial data. Recently, Song et al.[83] published their egocentric multimodal dataset recorded with smartglasses which contain egocentric video and inertial sensor data. In their work, they also showed their approach for recognizing life-logging activities. By utilizing Fisher Kernels they combine video and sensor features and reach high accuracy values. For our work, this approach may not suffice as it does not capture arm movement when it is out of frame. For the sake of completeness, we should mention that sensor fusion is also investigated in the field of robotics. But as it is mostly used for navigation and similar tasks, we do not consider work from that field.

5.3 DATASET

For this work, we created a dataset that covers order picking processes in a warehouse setting. In our previous work[30] (Section 3), we analyzed the impact of inertial data from a wrist-worn sensor on action detection. As this dataset puts less focus on the egocentric video, we created a new dataset that improves on that aspect. Initially, we have to define what actions make up a picking process in our dataset: Order picking consists of first looking at the shelf number, then walking up to the shelf, finding the correct box, picking an item from the box, looking at the item to simulate scanning it, and finally dropping it off at the start. In a real-world setting, these actions may vary slightly, depending on what existing technology is already in use. We recorded picking actions from four (three male and one female) participants, each performing 20 picking actions in two different settings. The following four cases were performed and recorded:

PICKING WITH THE ARM ACTIVITY FULLY IN FOCUS:
: In this scenario, the participants are focusing their view on the shelf while grabbing from a set of boxes. Half of the orders are from a shelf with boxes, the other half from an open shelf.

PICKING WITHOUT ARM ACTIVITY IN FRAME:
: Here, the participants were asked to specifically not focus on the shelf and instead look at something else. We had the participants look at the smartphone they are provided to emulate reading from an order list. Such scenarios are also likely to occur in a real warehouse environment, as experienced pickers often only glimpse at the shelf when working.

NO ACTIVITY WITH THE PARTICIPANTS LOOKING AT THE SHELF AND BOXES:
: Participants were asked to walk to the shelf with the intent of picking an item but without actually performing the grabbing action. We added this scenario as a negative example for our experiments.

NO ACTIVITY WITH THE PARTICIPANTS LOOKING AT THE SHELF AND MOVING THEIR ARM:
: This scenario serves a similar purpose as the previous one. But it adds arm movement (in the form of tacking out the smartphone from the pocket) as an additional action.

We recorded first-person view and inertial data with smartglasses and inertial data from a smartphone and a smartwatch. Additionally, all scenarios were filmed from a third-person perspective for improved labeling and easier validation of the actions. Figure 18 shows one participant with the devices and their on-body positions. The tablet was used to record depth data which may be used in future work. All inertial data was recorded using a mobile application from previous work[87]. Each inertial sensor was recording at a sampling rate of

50Hz. First-person video was collected at a resolution of 1920x1080 pixel with 24 frames per second. The smartwatch was worn on the right wrist, while the connected smartphone was kept in the pocket of the participants while recording. Our test environment consists of one shelf with multiple compartments. Each box or, in the case of open shelf picking, compartment has a unique QR code identifying the items. QR codes are ignored in this work but will be part of future analysis.

As the data was recorded with multiple devices, we first had to synchronize it. For this purpose, we introduced an *alignment motion* at the beginning of each recording. This motion produces a distinctive curve in the plot of the gyroscope data which we then used to calculate the time difference for each recording. We validate the difference by plotting inertial data of the watch and checking if the video timestamp overlaps correctly (see Figure 19).

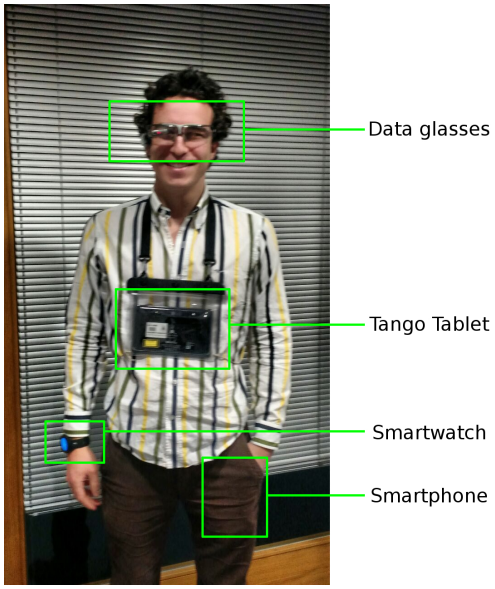


Figure 18: Participant wearing all devices for data gathering.

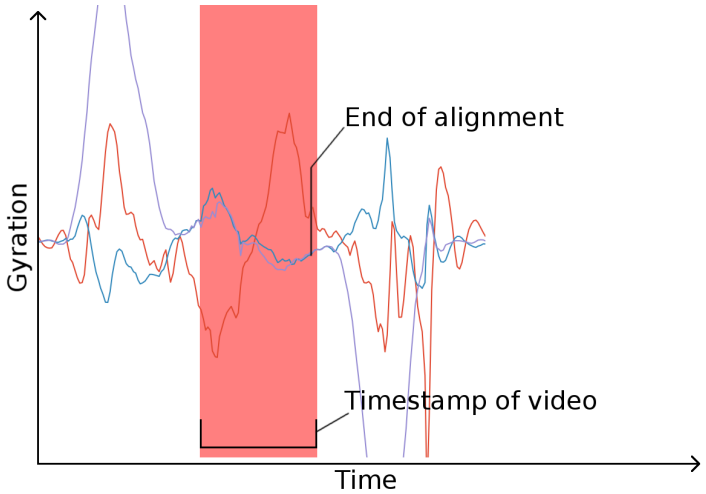


Figure 19: Plot of the alignment motion of the smartwatch with an overlay of the adjusted timestamp of the egocentric video.

After recording, the data was annotated two-folds: the first-person video and the third-person video were both labeled with the BORIS software[36]. The first-person video annotation includes the exact end of the alignment action, the timespan in which the hand is in frame while grabbing, and the timespan during scanning of an item. In the third-person video we also labeled the end of the alignment action and the whole grabbing process if present in the scenario. We plan to publish the data *.

5.4 METHODOLOGY

Time	**Frequency**
Mean, Variance, Correlation coefficient (Pearson), Gravity (pitch, roll), Standard Deviation, Median, Mean absolute deviation, Entropy (Shannon), Kurtosis, Interquartile Range (type R-5)	Energy (Fourier, Parseval), Entropy (Fourier), DC Mean
Color	**Texture**
HSV-Histogram, Mean of each channel, Standard Deviation of each channel	Histogram of oriented Gadients

Table 7: Features extracted from different modalities. Above the inertial features, below the image features. The recorded data was segmented into windows to compute inertial features where image features are computed on a per frame basis.

Our essential idea for learning grabbing actions is to leverage the combination of extracted features from inertial and video data. We consider features in the frequency and the time domain for inertial data as well as color and image descriptor features for the video data. Figure 20 shows the process of feature extraction and merging. For the frames, we extract histograms of the HSV color channels and histograms of oriented gradients (*HoG*[21]) (Figure 20, Step 1.1, 1.2, and 1.3). The histograms of the HSV channel are extracted without binning, enabling us to bin the data later. We also add the mean and standard deviation of each channel. The HoG features are generated with 25 patches per frame as a trade of between amounts of detail captured and feature size. All image features are extracted on a scaled-down version of the original frame. In total, this results in $(256 + 2) \cdot 3 + 25 \cdot 9 = 999$ features per frame.

For inertial data, we consider the acceleration data from the smartwatch. In a real-world scenario, we try to use the least amount of energy with the wearables, thus only acceleration was used. Inertial features are generated using a *sliding window* approach. This means, we consider a fixed timespan and calculate features on acceleration

* http://sensor.informatik.uni-mannheim.de

data within that span. Afterward, the window is moved to the next point in time, in the end resulting in a set of windows (Figure 20, Step 2.1, 2.2, and 2.3). Our features are calculated for a window size of 1000 milliseconds. This is a balance between too coarse window sizes for actions and windows without enough information in them. Consecutive windows overlap, allowing us to determine the start of a grab more precisely. We choose an overlap of 70%, resulting in 300 milliseconds between windows. This way, we can deal with the short actions we classify. Table 7 shows the features we calculated from the acceleration data of the smartwatch. These can broadly be classified into two groups: time-based features and frequency-based features. Furthermore, features can be sub-categorized on what property they are based on including distributions, shapes, and averages. Said properties will be used later to analyze subgroups within the feature sets. All inertial features are calculated on each of the axes of the acceleration data yielding $3 \cdot 14 = 42$ features for each window.

Since image features are calculated on a per frame basis and inertial features on windows, we have to combine them (Figure 20, Step 3.2). First, we have to align both feature sets with the alignment information we determined beforehand (Figure 20, Step 3.1). To merge the inertial and image features, we have to adapt the features extracted from the frames to fit the windows we calculated before. After we determined which windows a frame belongs to, we calculate the mean of each feature of all frames in every window, creating an *average frame*. As we store the labels of our dataset with the frames, we have to add that information to the windows. A window is thus labeled with the grabbing class if it contains at least one frame that also has this class. The combined windows are then stored per participant and scenario to enable different scenario combinations in our experiments (Figure 20, Step 3.2 and 3.3). By creating these combined windows, the evaluation of our experiments is more concise. Leaving either sensor out leaves us with the same windows, thus making a comparison between experiments easier. In the following, we are going to use machine learning algorithms on the combined dataset to see if our feature generation methods yield good results.

5.5 EXPERIMENTS

In the following, we present our experiments and their results in line with the research questions. First, we describe our experimental setup and subsequently conduct our experiments grouped by the research question.

5.5.1 *Experimental setup*

All experiments we conducted were tested with three classification algorithms: Support Vector Machine (SVM), Random Forest (RF), and Artificial Neural Networks (ANN). These algorithms were shown to work in similar settings in previous work[87][100][47]. Precision, Re-

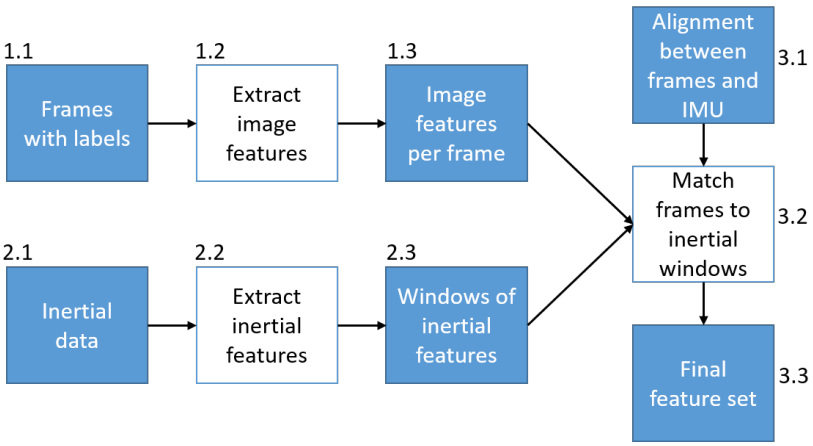


Figure 20: Process of feature extraction and combination. Steps 1.x and 2.x are happening simultaneously as they are independent of each other.

call and F_1-Measure of the classification are shown for each class separately with the measures for classifying the **grabbing** action being the focus in this work. Our whole dataset has a total of 8585 windows for non-grabbing actions and 1396 windows for grabbing actions. The classifiers used the following settings: A RF with a maximum of 100 trees and a depth of 10, a SVM-C with a polynomial kernel function, and a Multi-layer Perceptron with a maximal number of 500 iterations.

5.5.2 *Experiments*

To answer **RQ1**, we first apply the algorithms on the whole dataset with all features kept in place. We use 5-fold cross-validation with stratified sampling for the evaluation. Each algorithm is run 100 times with different folds to check if the results are stable. These results are shown in Table 8. It can be seen that the RF yields a high precision at the cost of recall while the SVM balances these values out. The ANN yields slightly worse results than the other two algorithms. This trend continues in subsequent experiments throughout this work. The results show that the combination of both modalities is very promising for recognizing the grabbing actions, with all numbers having a low standard deviation (SD). Still, we need to analyze how the classifiers perform within the timespan of a picking action. Our goal is to recognize a grabbing motion as early as possible, therefore we examine how well the start of an action is found. For this purpose, we look at the accuracy of the prediction in the first 100%, 75%, 50%, 25%, and 12.5% of all the windows of grabbing actions. Table 9 shows the results for our four participants. It can be seen that the results vary among the classifiers and participants. This is due to the fact that all the participants are grabbing at different speeds and also look at the shelf at different angles. We can also see that the low recall of the RF in Table 8 is reflected in the accuracy of the grabbing windows.

Method	Class	Precision	Recall	F_1-score ± SD
SVM	None	0.977	0.974	0.976 ± 0.003
	Grabbing	**0.845**	**0.862**	**0.853 ± 0.017**
	Average	0.959	0.958	0.959 ± 0.005
RF	None	0.956	0.995	0.975 ± 0.002
	Grabbing	**0.956**	**0.720**	**0.821 ± 0.019**
	Average	0.956	0.956	0.953 ± 0.005
ANN	None	0.962	0.956	0.958 ± 0.015
	Grabbing	**0.775**	**0.761**	**0.751 ± 0.054**
	Average	0.936	0.929	0.929 ± 0.019

Table 8: **RQ1**: Recognition quality of the actions grabbing vs. non-grabbing. All features are tested with 5-fold cross-validation and 100 runs on all data.

Generally, we have the highest accuracy in the first 75% of the grabbing windows. This is most likely due to the fact that participants are looking downwards at the end of a motion, not focusing on the shelf which removes prominent image descriptors. Accuracy in the first 12.5% of the relevant windows drops to the lowest value. Since grabbing motions start when the arm moves towards the shelf, and participants are likely to not focus on the shelf yet, determining the correct start is hard. In addition, the varying speed of the participants' grabbing actions poses a problem. Fast grabbing actions result in short sequences of grabbing windows. Thus, one misclassified window in the 12.5% subset has a greater impact on the accuracy value. This could be addressed by having a classifier consider the class of the previous window for its guess. For further analysis, we focus in the next experiments on feature subsets to explore their influence on classification results.

To answer **RQ2**, we analyze the influence of different features on the recognition rate. First, we split up the image and inertial features and evaluate them separately. The results are shown in Table 10. For the inertial data, it can be seen that among all algorithms precision and recall are dropping significantly. Results for image features though are much in line with the results of all features, though the recall drops for the SVM and the RF. From these results, we conclude that image features make up a significant part of the recognition. Still, the results of the combination of inertial and image features (Table 9) yields overall better numbers. We further analyze feature subgroups from the inertial data to find out if there are subsets of features that give us similar results to all inertial features. For this purpose, we create five feature subsets which can be seen in Table 11. Groups are created based on their domain, what they are representing, and on preliminary experiments. Table 11 shows the results of our feature subgroup analysis. We see that gravity by itself yields very good results. This is due to the fact that gravity consists of pitch and roll, thus it contains the relative position of the smartwatch. With partici-

Method	Participant	100%	75%	50%	25%	12.5%
SVM	P1	0.851	0.883	0.845	0.744	0.574
	P2	0.858	0.887	0.857	0.776	0.625
	P3	0.875	0.900	0.869	0.792	0.607
	P4	0.852	0.874	0.864	0.820	0.627
RF	P1	0.640	0.643	0.589	0.473	0.314
	P2	0.593	0.696	0.647	0.551	0.378
	P3	0.797	0.880	0.892	0.793	0.586
	P4	0.695	0.713	0.681	0.551	0.357
ANN	P1	0.681	0.706	0.662	0.564	0.405
	P2	0.761	0.803	0.759	0.687	0.548
	P3	0.803	0.839	0.798	0.658	0.450
	P4	0.753	0.773	0.725	0.632	0.533

Table 9: **RQ1**: Accuracy of all grabbing actions per participant in the first 100%, 75%, 50%, 25% and 12.5% of each set of grabbing windows.

pants grabbing from the same shelves, the position of the smartwatch can be used to register the arm's movement towards the height of the shelf. Since shelves in warehouses are rarely located on different heights (to minimize unergonomic movement) gravity can be a good indicator for a grabbing action. Drawbacks in this approach are varying heights of people, and arm movements that are similar to a grabbing motion. While height variation can be compensated with a bigger dataset, similar arm movement has to be recognized by other features. Features from the time domain are performing similarly to the gravity feature. As gravity is part of the time domain features, the good performance may be attributed to it. Still, the precision of all classification results improves when the whole domain is considered. The rest of our features perform worse, especially regarding the recall. It can therefore be seen that features from the time domain yield the best results for the task of grabbing recognition. This is due to the fact that our window size is smaller than the usual window size used for activity recognition. Since each participant performs the grabbing at different speeds and with different movements, the acceleration data by itself may not be sufficient for recognizing the action. Adding gyroscope and magnetic field information may improve the results.
In addition, we also analyze the image features (Table 10). Image features yield results close to the combination of all features. To rule out an overfitting of the data towards the QR code for instance, we analyze how the classifiers behaves in non grabbing scenarios. We evaluated how often the algorithms classified non-grabbing windows as grabbing windows in these negative scenarios. We found out that on average 2.1% of the windows in non-grabbing scenarios are labeled as grabbing actions. In a standard scenario grabbing actions make up roughly 10% of the windows. Therefore we can rule out overfitting on the QR codes.

Method	Features	Class	Precision	Recall	F_1-score ± SD
SVM	Inertial	None	0.902	0.983	0.941 ± 0.002
		Grabbing	**0.765**	**0.342**	**0.472 ± 0.023**
		Average	0.883	0.893	0.875 ± 0.005
	Image	None	0.949	0.994	0.971 ± 0.002
		Grabbing	**0.947**	**0.673**	**0.787 ± 0.018**
		Average	0.949	0.949	0.945 ± 0.004
RF	Inertial	None	0.923	0.978	0.950 ± 0.003
		Grabbing	**0.785**	**0.501**	**0.611 ± 0.025**
		Average	0.904	0.911	0.902 ± 0.006
	Image	None	0.943	0.992	0.967 ± 0.002
		Grabbing	**0.992**	**0.629**	**0.750 ± 0.019**
		Average	0.941	0.942	0.937 ± 0.004
ANN	Inertial	None	0.913	0.935	0.923 ± 0.010
		Grabbing	**0.549**	**0.448**	**0.478 ± 0.055**
		Average	0.862	0.867	0.861 ± 0.009
	Image	None	0.957	0.959	0.957 ± 0.016
		Grabbing	**0.779**	**0.732**	**0.737 ± 0.061**
		Average	0.932	0.927	0.926 ± 0.020

Table 10: **RQ2:** Separate analysis of inertial and image feature sets concerning the recognition quality. The experiments are conducted in context of 5-fold cross-validation and 100 runs.

After the feature subgroup analysis, we further evaluate the performance of the classifiers for the start of the action. For this purpose, we again evaluate the accuracy of the algorithms for the first 100%, 75%, %25, and 12.5% of windows of all grabbing windows. Table 12 shows the results of this experiment. While the overall performance is in line with the feature experiments in Table 10, the performance for the different percentages differs greatly. It can be seen, that the accuracy varies stronger for the different participants when compared to the results in Table 9. This fact can be explained with arm movements having greater variation compared to the frames of the participants. In addition, the acceleration sensor is worn on the wrist, therefore it is prone to noise. The arm of the participant is always moving while recording and even slight motions while walking pose a challenge to the classification algorithms.

Overall, we can see that a combination of inertial and video data can return fairly good results for grabbing classification. Image information is much more valuable for a precise classification, but can be enhanced with inertial data for better recall. Inertial data by itself though, when only measured at one position, is too noisy to easily identify a grabbing action. This is due to the fact that the watch moves on the participants' wrist.

Method	Features	Precision	Recall	F_1-score ± SD
SVM	Mean, SD, Var	0.697	0.099	0.173 ± 0.022
	Gravity	0.625	0.264	0.369 ± 0.033
	MAD, IQR, SD, Var	0.626	0.029	0.054 ± 0.014
	Time	0.739	0.302	0.429 ± 0.024
	Frequency	0.647	0.077	0.134 ± 0.021
RF	Mean, SD, Var	0.594	0.258	0.359 ± 0.022
	Gravity	0.652	0.474	0.548 ± 0.021
	MAD, IQR, SD, Var	0.506	0.134	0.211 ± 0.024
	Time	0.765	0.444	0.562 ± 0.022
	Frequency	0.607	0.251	0.354 ± 0.025
ANN	Mean, SD, Var	0.640	0.222	0.328 ± 0.037
	Gravity	0.639	0.344	0.445 ± 0.038
	MAD, IQR, SD, Var	0.586	0.076	0.132 ± 0.041
	Time	0.701	0.530	0.599 ± 0.037
	Frequency	0.476	0.291	0.338 ± 0.078

Table 11: **RQ2**: Different subsets of inertial-based features, analyzed in context of the action **grabbing**. The experiments are conducted in context of 5-fold cross-validation and 100 runs.

5.6 CONCLUSION

We could see in our experiments that the combination of both modalities outperforms every single modality when trying to detect grabbing actions. For **RQ1** we are able to show that combining the sensors to balance out the drawbacks of each one yields an F_1-Measure of **85.3%**. However, finding the correct start of an action can still be challenging and would need to be further investigated in future work. Improvements could be done by weighting the start of an action greater than the rest of the actions, therefore creating a classifier focused on finding action starts. In addition, data from the gyroscope could also be used to get a more robust classifier that may be more precise in its prediction. Since the start of the grabbing action often includes rotations of the wrist, this approach could lead to greater results in finding the start of the action.

The feature analysis in **RQ2** shows that the combination of both sensors led to the most promising results. It also can be seen that for short actions inertial features from the time domain work better than features from the frequency domain. Future work will focus on two main topics: First, we want to explore the usage of more inertial data. Currently only the inertial data from the smartwatch is analyzed in our approach. By also considering inertial data from the smartphone, we can get a better notion of the subjects' movements. This way we may leverage the fact that a person is standing still while grabbing

Method	Participant	100%	75%	50%	25%	12.5%
SVM	P1	0.383	0.433	0.323	0.170	0.214
	P2	0.281	0.335	0.380	0.323	0.290
	P3	0.473	0.513	0.522	0.530	0.440
	P4	0.237	0.246	0.174	0.087	0.088
RF	P1	0.488	0.510	0.423	0.236	0.198
	P2	0.378	0.434	0.463	0.452	0.437
	P3	0.620	0.672	0.696	0.655	0.581
	P4	0.280	0.247	0.149	0.094	0.070
ANN	P1	0.507	0.511	0.486	0.423	0.380
	P2	0.433	0.479	0.487	0.529	0.516
	P3	0.572	0.597	0.591	0.597	0.613
	P4	0.334	0.307	0.264	0.211	0.255

Table 12: Accuracy of all grabbing actions per participant in the first 100%, 75%, 50%, 25% and 12.5% of each set of grabbing windows for **inertial features**.

from a shelf. The second topic we want to explore is a better merging of inertial and video data in addition to more elaborate video features. On top of calculating an *average frame* for each window, more complex methods like object detection with neural networks could be used for an overall more robust approach. From these features, we may explore different fusion approaches, e.g. early versus late fusion.

6

PREDICTING KITCHEN ACTIVITIES IN MULTIMODAL SETTINGS

In the previous chapters, we have seen how data for Human Activity Recognition can be annotated easily and the challenges researchers face when they try to segment data into sequences of activities. Now, we try a multimodal approach of recognizing activities on the aforementioned CMU-MMAC dataset.

6.1 INTRODUCTION

In the field of pervasive computing, many researchers suggested solutions for the task of human activity recognition [1, 70, 82, 87]. One popular task of this field is the recognition of so-called activities of daily living [53]. As the cost for care increases [4, 41, 94], many fields in the area of health care and nursing could benefit from computer-aided solutions that support caregivers. One computer-aided solution that is often suggested is the use of smart home environments. Here, activities of the patients or the people in need of care are inferred from sensors that are installed in the living area. However, these approaches can be very costly, as they often have to be adapted to each environment separately and may require a relatively big infrastructure to work properly. Additionally, the task of recognizing individual activities gets harder if a person is sharing their living space, as sensor events can be attributed to multiple people, making distinguishing the events difficult. In recent years, the market for smart devices has grown significantly with devices such as smartphones, fitness trackers, smartglasses, and more becoming more easily available for consumers. Hence, researchers now have a great pool of possible sensors to use in a multi-sensor system. We propose the usage of such off-the-shelf smartdevices to recognize the aforementioned activities, where we rely on inertial sensors and an egocentric camera for our prediction.

Several studies already investigate activity recognition, be it low-level [69, 87, 92] or high-level activities [73, 78, 82]. Usually, the former consist of actions like *standing* or *walking*, whereas the latter refer to context-enriched actions such as *preparing food*. To recognize the latter, researchers propose the use of head-mounted cameras built into devices like smartglasses. Their results show that object-based activity recognition is one of the most promising vision-based approaches [13]. However, the object recognition itself is error-prone and at the same time crucial in respect to the recognition quality [73] therefore making it a vital part of the whole pipeline. Smartphones and smartwatches that are equipped with accelerometers, gyroscopes, and magnetometers are another popular choice. In contrast, inertial-based high-level activity recognition approaches usually perform less

accurate but are a reliable option for low-level activities. This also includes the tracking of the user's arm [92] which we need for our approach. Therefore, many researchers started to adapt the approach of fusing multiple sensors to get a better overall result. Approaches for fusing inertial and vision sensors have been made by other researchers already [84]. However, most of the work focuses on the fusion of sensor streams that belong to the same on-body position [82, 95] and rarely looks at different body positions. Complex activities have often been detected using smart environments with sensors attached to objects and location to recognize interactions. Such approaches can give exact results regarding the interaction with an object but are expensive to deploy in real-world scenarios given the high amount of variation in home environments.

This chapter reflects the content of our previous publications [34][32][29].

6.2 RELATED WORK

There are several methods and publications from the domains of image and video processing that target subproblems of our research question. Similarly, using inertial data for activity recognition has also been researched in depth. The approaches in both of these fields have shown to perform well in their respective applications. In the following, we summarize methods that can be used to support multimodal activity recognition. Namely, we first look at separate methods for vision and inertial data. Afterwards, we consider research for combining both of them.

6.2.1 *Image Object Detection*

In recent years, there have been advances in deep and neural network based object detection. One prominent example is the TensorFlow Object Detection API[1], that integrates many popular architectures in one easy to use API. The API offers deep learning based approaches for object detection that rely on pre-trained models which were initially evaluated on the Microsoft COCO object detection challenge [43, 57]. Given an image, the TensorFlow model generates bounding boxes for potential objects and annotates them with object classes. Each annotation is associated with a confidence value, allowing users to work in-depth with the data. Thus, we decided to use this framework for our methods. Many different neural network architectures are offered and have their separate advantages. One typical trade-off is between performance and run-time. A currently well-performing network is NASNet with a Faster-RCNN [103] which yields an mAP score of 43.1%. In our case, we rely on a ResNet FPN model as described in [56], as the reported performance of 35% mAP is still among the best offered. However, it offers the advantage of a run-time that is significantly lower than the state of the art network (1833ms vs. 76ms).

1 https://github.com/tensorflow/models/tree/master/object_detection

Using these object information, we can work towards recognizing activities.

6.2.2 *Activity Recognition Based on Objects*

Using object information for activity recognition, especially when looking at complex activities like cooking for example, has been explored by many researchers [22, 50, 88]. For this purpose, the occurrence of objects and possibly also the interaction with said objects is used to recognize an activity. Wu et al. [99] already showed good results by detecting changes in objects positions, using an RFID sensor as a way to validate the interaction. In this case, the camera was stationary, pointing towards the location of the actions, thus making the detection of change a feasible approach. Similarly, Lei et al. [54] build their system on a RGB-D camera system, detecting activities in a kitchen environment. Here, the focus was put on the recognition of actions and objects, utilizing tracking methods, and object detection. Adding a camera to a wrist-worn sensor is another approach for detecting activities and was analyzed by Lei et al. [60]. A wrist-worn camera has the added benefit of having interactions with objects always in frame. Also, the hand movement is synchronized with the camera movement, making reasoning about egomotion vs. outside movement easier. Recently, Kumar et al. [50] used off-the-shelf object detection network and transfer learning to find correlations between predicted object labels and ground truth data of activities. This approach is very promising, as it explores the transfer of deep learning models in vision to models for activity recognition. One problem image-based recognition models face in practice is a limited field of view of the camera. When an activity occurs that is not fully captured within the field of view, the information is lost to a system. Systems that use stationary cameras may not suffer too much from this issue but involve an initial setup of a smart environment and are less flexible in their usage. Therefore, we additionally look at inertial data which also has been used by many researchers to detect and recognize human activities.

6.2.3 *Activity Recognition Based on Inertial Data*

One main reason for the increased focus on inertial data for activity recognition is the rise in popularity of smartdevices that often have a series of sensors (including inertial sensors) built into them. In this context, inertial data typically refers to acceleration, gyration and magnetic field data. Sliding windows in combination with acceleration data are a typical method to predict activities and have been analyzed by many researchers before [51, 74, 81]. Especially activities like walking, jogging, and climbing stairs have been predicted successfully. Hereby, the position of the acceleration sensor is one important factor that has been considered for the prediction [74]. Sensors are often placed on the legs, the arms, and the torso of subjects and

then evaluated either separately or in combination. Features that are calculated from these windows are often from the time and frequency domain and may consist of measures like mean and variance but also more computationally expensive and complicated features like energy [87]. Apart from cyclic activities, researchers also use inertial data to detect short activities or events. A common use case for short activities is the detection of accidents like falls [28, 49, 91]. As our scenario also involves many short activities, these methods are interesting to our problem setting. Falling, however, is an activity with a unique motion that is hard to mix up with other activities of everyday living. Therefore, we cannot fully utilize the methods presented there and have to adapt them to our needs. Algorithms that are commonly used for classification in this field are Decision Trees[51], Hidden Markov Models[81], and kNN[74]. But recently, Ordóñez et al.[71] also employed neural networks for similar tasks. Here, it has been shown that by adding new modalities (for example gyration data on top of acceleration data) to a network, features can be extracted automatically without the need for manual pre-processing. By using convolutional layers in their network architecture, every added modality was adapted properly without the need for manual feature engineering. In our work, we rely on a sliding window approach, similar to [87]. But in contrast to low-level activities, where the window size can be fairly long, thus capturing abstract characteristics of an activity and better dealing with noise, we rely on short windows with a high overlap between consecutive windows. This way we aim to capture the short nature of our activities within the windows while also allowing for an easier fusion later. We looked at the separate methods for activity recognition using video and inertial data and are now examining methods for fusing them.

6.2.4 *Multimodal Activity Recognition*

Previous work presents multiple methods to combine sensors and to create and analyze multimodal datasets [19, 83, 88]. Scenarios recorded in the datasets vary greatly and involve activities such as office work [83], sport activities [19], and cooking [88]. Using these datasets, researchers developed and evaluated different methods to recognize activities. Some datasets [18] use a multimodal approach that combines ambient and wearable sensors. Spriggs et al. [84] for instance, fused vision and inertial data to recognize cooking activities. One problem that is central in dealing with multimodal datasets is the fusion of sensors with different sampling rates. Inertial data is usually sampled at a higher rate than video data, especially when using off-the-shelf sensors. Spriggs et al. [84] solved this problem by downsampling the inertial data to the capture rate of the video, thus having a one-to-one mapping of frames to single inertial measurements. When dealing with windowed feature, some of these problems can be mitigated. By defining windows via start and end time rather than number of instances, one central timeline for data from

different sources can be used. This allows for an easier merging of the different modalities. Once a valid temporal mapping is available, the problem of fusion methods can be addressed. Song et al.[83] published their egocentric multimodal dataset which contains video and inertial data from smartglasses. To recognize life-logging activities, they developed and presented a fusion method for inertial and video data. They combine the modalities with Fisher Kernels and could reach a high level of accuracy. Other methods of fusing multiple modalities often rely on Kalman Filters [6, 46], where the results are often used in the fields of robotics. In these scenarios, however, the camera and the inertial sensors are located at the same place. Thus, both sensors capture the same motion. Our scenario has the inertial measurement unit capture the movement of the arm, while the camera is located on the subjects head, thus such fusion techniques may not be easily applicable.

Multimodal activity recognition can also involve the combination of stationary and wearable sensors. De et al. [25] has shown such an approach for the healthcare sector. Here, both multipositional personal sensors and the combination of ambient and wearable sensors have been utilized. For multipositional approaches, results use classification outputs of each sensor separately and yield an improvement over single sensor usage. Combining ambient and wearable sensors allowed for better classification in regards to location-dependent activities e.g. opening a fridge or lying in bed. For complete stationary sensors, one recent example was authored by Zou et al. [104] who utilized stationary cameras and WiFi signals to recognize locomotive activities. Using late fusion, deep learning models are learned for each modality separately and finally combined with different ensemble methods.

Radu et al. [75] explored deep learning for multimodal activity recognition. Specifically, a RBM architecture has been applied, outperforming shallow classifiers. For modalities, they used a public dataset [85] that compares different sensors in smartphones and smartwatches. Similarly, Guo et al. [40] utilized sensor ensembles for multimodal activity recognition based on a neural network architecture. The method combines heart rate signals with data from IMUs, with a focus on ensemble learning. Similar to our approach, the models are learned for each modality separately and finally combined with a meta-learner. Combining vision and inertial data on a feature level has also been proposed by Ehatisham-Ul-Haq et al. [35]. Here, HoG features are used for RGB-D data and time domain based features for inertial data. Features are fused early and used to train kNN and SVM classifiers. Roitberg et al. [79] showed a multimodal approach with multiple depth-enabled cameras for industrial manufacturing. To calculate meaningful features, they captured skeleton information from the sensors as well as the motion of specific joints. As the cameras capture similar information, PCA was applied to reduce the feature space. Depth and RGB data was also combined by Wu et al. [98] for gesture segmentation and recognition using neural networks. A comparison of early and intermediate fusion was made based on the

layer within the network the data was combined. Late fusion outperforms the results of earlier concatenation of the data. Radu et al. [76] presented a deep multimodal approach that works for different types of sensors, e.g. multiple inertial or multiple EEG sensors. For that approach, they tested against shallow learning methods and could show good results in most scenarios with their CNN and DNN based methods. This shows, that similar architectures can work for different modalities. In this work, we rely on a windowing approach for both of our modalities. By aligning them and then using windows defined by a timespan, our data can be merged and we can evaluate late and early fusion approaches for our task.

Especially with goods like food, placing such sensors may not be feasible on a bigger scale. This can be seen in practice in the currently tested retail shops created by Amazon [44] where good results are achieved but the amount of sensors needed is very high. In addition, an interaction with an object that was registered via a sensor may not translate to a properly performed activity (for example if a pill box was touched by a user but no medicine was consumed).

We present our work on a multimodal egocentric activity recognition approach that relies on smartwatches and smartglasses to recognize high-level activities like activities of daily living. For that purpose, we combine inertial and video information and try to take advantage of each of their strengths. Particularly, we consider the inertial data of our smartwatch to classify the movement pattern of the forearm. The video data provides object information from the smartglasses. We aim to investigate to what extend vision information can improve the recognition of activities that are hard to recognize purely through motion sensing. This is especially the case when motions are short or very similar (e.g., eating vs. taking medicine). We present the results of our multimodal activity recognition approach based on manually annotated video data. In addition, we test our approach on a public dataset that contains data from similar sensors but set in another scenario. Specifically, we look at the CMU-MMAC dataset [88] that contains recordings of people cooking different recipes. Our contributions in this work are:

1. We collected a new dataset with two subjects performing a set of activities in two different environments with a focus on activities that are hard to distinguish as they involve similar motions (e.g. eating and drinking) and are often interleaved. Each subject performed the activities in different human body positions and at different speeds. Currently there are few datasets that cover these scenarios thus other researchers in the field can test their approaches on this dataset.

2. We present a new method and a baseline comparison for multimodal activity recognition, utilizing deep learning models for object detection and evaluating this method on our presented dataset, achieving an F_1-measure of 79.6%. We also apply our method to the CMU-MMAC [88] dataset and can show that we outperform previous work on the same dataset. Additionally,

we test our method with a greater subset of the CMU-MMAC dataset, as a recent publication offers more annotations [101].

6.3 DATASET

In this work, we look at two separate datasets to test and evaluate our developed methods. The first dataset was collected by us and deals with a subset of activities of daily living. It focuses on activities that are hard to distinguish just based on the inertial data, as they involve very similar motions. The second dataset we looked at is the CMU-MMAC dataset which contains a wider variety of activities with more test subjects. Namely, the dataset has recordings of people preparing different recipes in a test kitchen environment. In the next two subsections, both our dataset and the CMU-MMAC dataset are described in regards to content, size, and target classes. We also describe a new set of annotations for the CMU-MMAC dataset that has been published recently. For our own dataset, we also go into detail about the recording process of the data.

6.3.1 *ADL Dataset*

For this dataset, we recorded two test subjects performing different activities in a typical home environment. All the recordings have been done in an experimental setting and the test subjects consented to have the data recorded and published. Furthermore, we remove the audio track from all video recordings and cut away all video data that was not part of the scenario. The egocentric video was obtained from two angles: via smartglasses and a chest-mounted tablet. Additionally, we recorded the test subject from a third-person view and used the video for the annotation of activities. The subjects were also equipped with smartwatches and smartphones to capture the movement of their arms and thighs. Here, we recorded acceleration, gyration, and magnetic field data for all sensors simultaneously. This way, we only focus on the scenario and leave out any conversations or other interactions of the subjects within the test environment. The subjects performed common and interleaved activities which include *drinking* (A_1), *eating* (A_2), *taking medicine* (A_3), *preparing meal* (A_4), *taking snack* (A_5), and *wiping mouth* (A_6).

The procedure of the recording sessions was predefined, as the whole dataset is not too big in size and variation would make proper classification unfeasible. Hence, the subjects executed two certain sequences (A_5, A_3, A_1, A_5, A_6 and A_4, A_2, A_3, A_1, A_2, A_6) where each sequence was performed two times. Once, it was done in a natural fashion and the other time with short interruptions between the individual activities. This way, we have scenarios where the activities are easy to separate and others where they are slightly overlapping. As these activities can be performed in several different postures, i.e., standing, sitting, and partly also lying, we recorded several sessions for each posture separately. To add more complexity, the home envi-

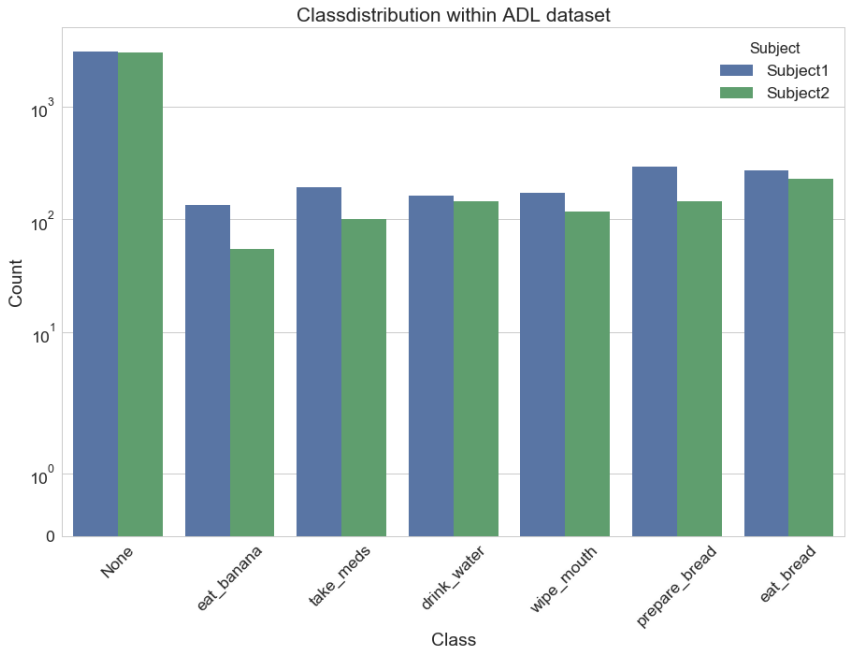


Figure 21: Distribution of classes per test subject using logarithmic scale as the majority of class labels belong to the none class. It can be seen that the majority class (excluding the none class) changes for each subject.

ronment of the recordings switches between two different locations, adding different backgrounds to the video. Overall, we recorded six sessions per subject which results in 30 minutes of activities. Figure 21 shows the distribution of classes among the subjects in the subset of sitting activities. The difference in distribution can be attributed to differences in performing an activity, where *subject1* for example was taking more time to butter their bread than *subject2*.

The required data was collected using different smart devices[2] (see Figure 23) which were attached to the head (P_1), the left (P_2) and the right (P_3) wrist, the chest (P_4), and also to the left (P_5) and right (P_6) thigh. Video and inertial data was recorded with a resolution of 1920x1080 (25fps) and 50Hz, respectively. In this context, the parameters were chosen with reference to related studies [51, 70]. Data was collected via an app that we developed(see Figure 22). Each device was running an instance of the app and stores the data it receives in a local SQLite database. Different sensors like temperature, audio level, and others can also be captured via the application presuming the device has said sensors built into it. The binary and the source code[3] for the recording application are publicly available.

After the recording, we manually annotated the collected data on an activity level defined via start and stop time and the performed activity. Annotations are based on the third-person recording of the data as this view fully captures the motion of the test subject. Egocen-

2 "Vuzix M100" (Glasses), "LG G Watch R" (Watch), "Tango" (Tablet), "Samsung Galaxy S4" (Phone)

3 `https://sensor.informatik.uni-mannheim.de/#collector`

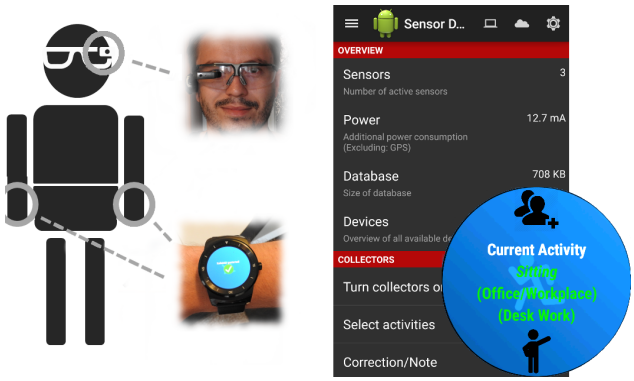


Figure 22: Sensor data collector application. The application is able to record a big set of sensors in Android devices including inertial data, temperature, and audio for example.

Figure 23: Sensor placement. The subject wears the wearable devices on the head, chest, forearm, and thigh (top-down).

tric vision can often leave out the proper start and end of an activity, as the field of view of the camera does not allow to fully capture all the movement. For labeling the activities we used the *Behavioral Observation Research Interactive Software* [36]. On an object level, we drew the required bounding boxes around the visible objects within the egocentric video of the smartglasses. In this context, we marked 14 objects including *bread*, *napkin*, *glass*, *knife*, *pillbox*, and both hands. Figure 24 shows an example of the bounding boxes and also highlights that most objects are usually blurred or partly out of frame. Labeling of bounding boxes was done with *vatic* [90].

Data that was recorded on the same device (i.e. the smartwatch paired to the smartphone) uses the same clock (namely the internal clock of the android phone) and thus does not to be aligned. But to further work with the data, we had to align all sources of data to be able to work within one consistent time-space. To be able to align the data easily, the test subjects started the recording with a period of no movement. This way, we could pinpoint the start of the movement for each sensor and therefore could calculate the time

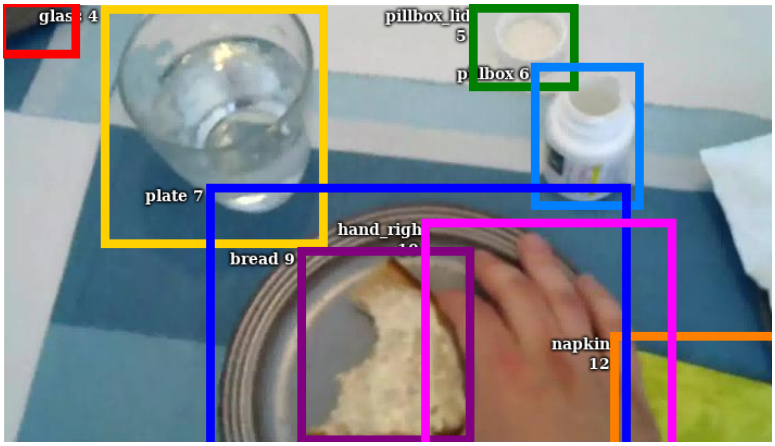


Figure 24: Example bounding boxes. It depicts a usual frame that was captured by our smartglasses. We draw the bounding box for each object, even if it was only partly visible. The boxes were tagged concerning the visibility state of the object.

difference among them. We annotated the start of the motion with the *boris* annotation software for both the egocentric and third-person video. Simultaneously, we mark the same point in time in the plot of the acceleration data and store the resulting timestamp. Using the alignment points, the activity labels can be mapped to any of the collected sensor data. This means we can assign each frame of a video a timestamp that is consistent with the timestamps of the acceleration data.

Our labeled dataset is publicly available, including a detailed description and images of each subject and the environment[4]. In this work, we rely only on the smartglasses and the smartwatches. With such a setup, we try to maximize the recognition performance but still use a fairly small amount of sensors.

6.3.2 *CMU-MMAC - Quality of Life Dataset*

The Quality of Life dataset [88] was created by the Carnegie Mellon University and contains a large set of test subjects, cooking a total of five different recipes. Modalities that were recorded include first-person overhead video, inertial measurement units that record acceleration, gyration, and magnetic field data on different body positions, audio from five different microphones, and in some cases even motion capturing data. With the recording of so many different sensors, synchronization becomes an issue. The authors have used two different methods to address this challenge. First, the recordings of the sensors we consider in our experiment (video and inertial data) have been done centrally on one laptop that the subject is carrying with them. This way, single frames and readings of the inertial data are synchronized on one device, as they use the time of the laptop. For the rest of the data, synchronization among the devices has been

4 https://sensor.informatik.uni-mannheim.de/#dataset_egocentric

achieved by synchronizing the clocks on all computers with the NTP protocol.

For our main analysis, we focus on a subset of recipes, the brownie recipe, as labels for these recordings are provided by the original authors. We use this dataset to analyze different challenges and questions that cannot be addressed in our own dataset. One question is the behavior of our model when trained on a larger dataset. As we only have two test subjects in our recordings, we want to use the CMU dataset to test our method on a bigger dataset. The subset of the Quality of Life dataset contains thirteen different test subjects compared to our two different test subjects. This yields more variation, as more subjects are performing the activities and in total also more data is available to train and evaluate our model. Another challenge is the complexity of the labels which is already obvious due to the more complex scenario of cooking. Annotations are given in the form of *verb-object1-preposition-object2*. Here, the brownie recipe consists of 17 different verbs, 34 different objects and 6 different prepositions. Overall we counted 43 different labels in the subset we considered. With so many classes, and some of them only having few instances, a learned model would overfit to these instances resulting in biased numbers. Additionally, the combinations of verbs, objects, and prepositions can become very big (when all combinations are considered as a possible target class) and our proposed method targets a closed set of activities (e.g. taking medicine in the ADL scenario). Thus, we decided to group the labels in some form to be able to create a meaningful model. To achieve this, we only look at the verb part of the activity as our target class. While vision information is needed to determine the objects used in the activity, only the verb part really benefits from both inertial and vision data. Our assumption is that some activities with the same verb share common movement patterns but only in combination with the vision information we can distinguish some classes. This reduces the number of classes to 14 and also allows us to compare our method to previous methods like [102] who also used only the verb part of the activities. Figure 25 shows the distribution of the different classes in the dataset that we consider in our work.

In total we looked at 13 different subjects, only considering the overhead camera frames and the acceleration data on both arms in our analysis. Data was aligned and trimmed with the provided synchronization files, and afterwards cut to the length of the sequence of activities.

6.3.3 *CMU-MMAC - New Annotations*

Recently, a new set of annotations for the CMU-MMAC dataset was released that vastly increased the number of labeled scenarios. In [101], the authors showed their approach for annotating the data while also offering semantic annotations that can be used in other experiments, e.g. when utilizing reasoning. Overall, they added annotations for three recipes and for all subjects, with the exception of

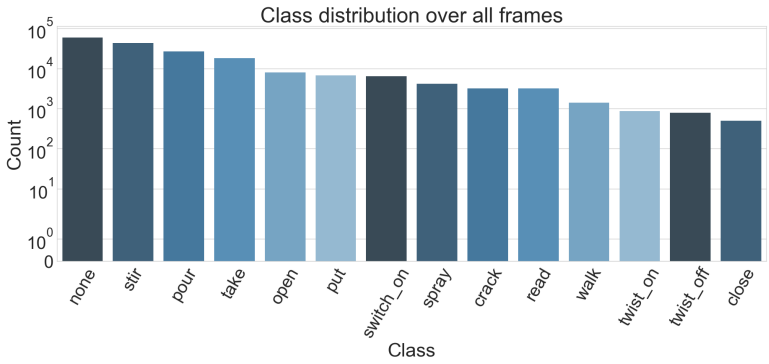


Figure 25: Distribution of the classes we consider from the CMU-MMAC dataset. The class label is derived from the verb part of the original label.

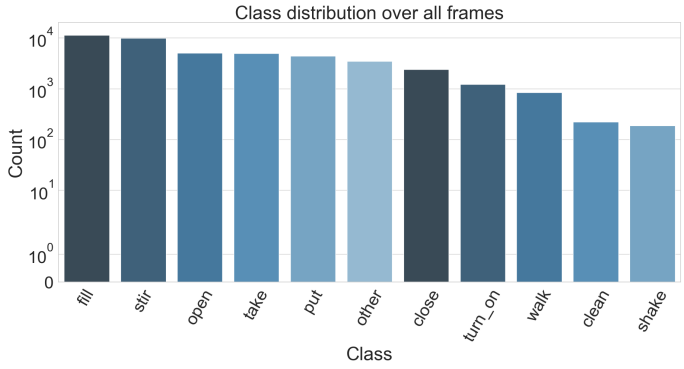


Figure 26: Distribution of the classes we consider from the CMU-MMAC dataset using the annotations from [101]. The class label is derived from the verb part of the original label.

cases where the video files were broken and could not be used. Annotations are mostly based on the first-person view, thus making it easy to use with our previous approach.

To make learning activities feasible, we were only considering one recipe: baking brownies. This way, we alleviate an issue with the annotations and our problem description. Labels are given in a similar fashion as they are on the official CMU-MMAC dataset website. Namely, they use the form *verb-object1-object2-...-object_n* to properly specify the activities, where the number of objects can vary depending on the scenario. An example would be the class *(open drawer)* vs. *(fill oil oil_bottle pan)*. However, this once again yields a huge number of different labels (in the subset of brownie recipe, there is a total of 165 different annotations) which in turn makes learning each one of them unfeasible especially since 59 of these labels have less than 10 instances in the dataset. Instead, we are also only considering the verb part of these annotations. When we look at all recipes, however, there are a lot of cases where a verb is used with objects that are unique for each recipe. This in turn makes the group of activities with the same verb very heterogeneous, thus making it difficult to learn the specifics of an activity. Therefore, to make the learning feasible and

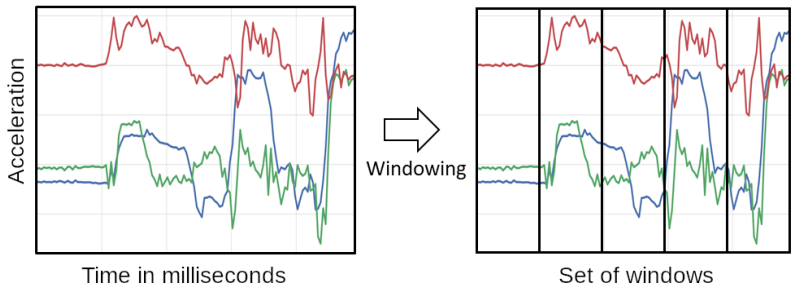


Figure 27: Windowing of inertial data. Windows have a length of 1s and an overlap of 50% or 75%.

also compare the results to the original annotations, we only look at the complete set of recordings for the brownie recipe. Similar to figure 25, figure 26 shows the class distribution of the verb labels in all the sequences for the brownie recipes.

6.4 METHODS

6.4.1 *Acceleration Data*

Time domain	Frequency domain
Mean, Median, Standard Deviation, Variance, Inter Quantil Range, MAD, Kurtosis, Correlation Coefficient, Gravity, Orientation, Entropy (Time)	Energy, Entropy (Frequency), MeanDC

Table 13: Set of features from acceleration data. Features are in the time and frequency domain.

To keep the number of used sensors minimal, we only consider acceleration data from the smartwatches. Activities we aim to recognize are mostly performed with the hands, allowing us to only consider said wrist-worn sensors. Other inertial data that may be interesting for activity recognition in our scenario is the data collected by the smartglasses. One example for using the inertial data of smartglasses may be to give a better understanding of when a subject is moving their head to take a sip from a cup. Initially, we planned to only consider data from the dominant hand of the test subjects, but as activities were often performed with a mix of both hands, we decided to use both. For our features, we use a sliding window approach. Figure 27 visualizes the windowing of inertial data. To generate the windows we use a framework we developed that is publicly available[5]. Here, the framework first takes all data for one modality and

5 https://sensor.informatik.uni-mannheim.de/

calculates temporal windows based on a set of parameters it receives. Thus, we transform the time series into a set of discrete windows which allows us to analyze them separately. Temporal windows also have the advantage that they can contain different amounts of data points per window in contrast to typical windowing approaches which are defined by the number of points they contain. Defining the window size via a timespan makes the approach more robust in potential real-world settings, as sensors may drop single readings which would result in a shifting of the windows. After the windows are defined, data points are added to the windows and each window calculates a set of features (described in Table 13). Some of the features are in the time domain, others in the frequency domain. Generally, the prediction power of features can vary as we showed in a related work [33], but in this scenario, we kept all features and let the learning algorithm decide which ones to use. For the parameters of the framework, we set the length of the windows to 1000ms and overlaps of 50% or 75% (see Figure 27). We base our settings for the window size on previous works in the field [87] as well as adapting it to the scenario. Longer window sizes than 1000ms would not be feasible, as the activities we consider are too short and windows would contain multiple activities. Shorter window lengths, however, would not capture enough specifics about the movement to properly distinguish the activities. The overlap allows us to look at the dataset with a finer resolution which, given the short nature of some of the activities, can be very useful. For motions like raising an arm towards a glass or picking up items, inertial data may be sufficient. But to properly detect the different activities, we also have to consider the visual information. This is because acceleration information may not be able to differentiate between objects, e.g. in the cases when food or medicine is picked up.

6.4.2 *Video*

Video features in our model are based on object information within the frames. To recognize the objects, we use a pre-trained object detection neural network and transform its results into feature vectors. As described in Section 6.2, we use bounding boxes of a ResNet FPN network. Masks of the objects were also considered initially. But the added benefits of more details are outweighed by the significantly longer run times for detection and the comparison with our ground truth, which is present in bounding box format, not being ideal. When looking at the activities from the first-person view, we can see that a main component of the activity is the interaction of the test subject with different objects. We assume that interactions with different objects are a good indicator for an activity and preliminary experiments verified this assumption. In these experiments, labeling the interaction with a video annotation tool, and using these interactions as a feature vector, we could show a very high performance of the model (close to 100% accuracy). It thus can be seen that

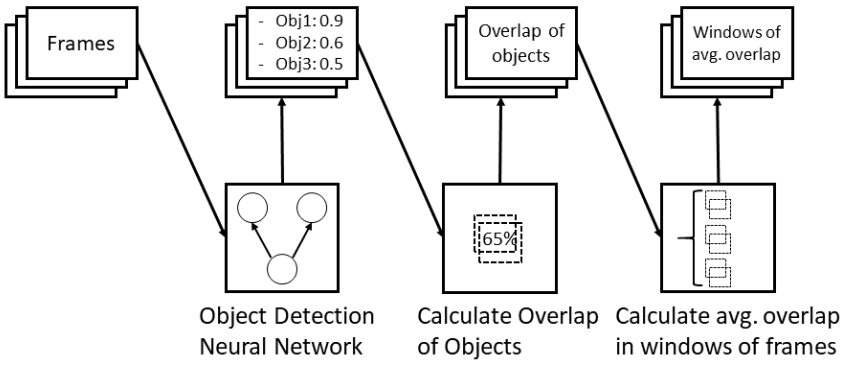


Figure 28: Pipeline for the image feature generation

recognizing interactions of a person with their environment should be one main goal of our approach. Our estimation of an interaction works by looking at the overlap of bounding boxes from a detected hand and any other type of object. Therefore, we first pre-filter the frames and only consider those that contain a positive detection for a hand (which is labeled as a *person* within the target-classes of our neural network). In these frames, we then calculate the overlap of each detected object's bounding box with the hand's bounding box. This results in a vector with the length equal to the number of object classes that can be detected by the neural network not counting the person class (as an overlap of the hand with itself does not add any information). The rest of the frames are assigned to a vector of the same dimension filled with negative ones as values for each interaction. For each frame, we thus get a feature vector that describes which objects are present in a frame and how much they overlap with the detected hand.

To further work with the generated image features (and especially combine them with the inertial data), we apply another windowing approach to them. Here, we consider a window of frames where we calculate the average overlap of each object with the hand within the window. For the window size, we choose a value of ten frames with a stride of five. We use windows, as within the video a hand often hovers over different objects when performing an activity. Therefore, overlaps are often calculated even though no interaction with any objects occurred.

We assume that interactions with objects yield a longer span of time where the detected hand overlaps with the object. Thus, the mean overlap within a window is greater for interactions than an overlap of the hand when only passing an object. The whole process of extracting vision features is described in Figure 28.

To evaluate the approach further, we run the experiments on the learned image features as well as on object annotation ground truth data. This way, we can analyze the reliability of the vision features without dealing with wrong or missing classifications from our object detection network. As we do not have object annotations for

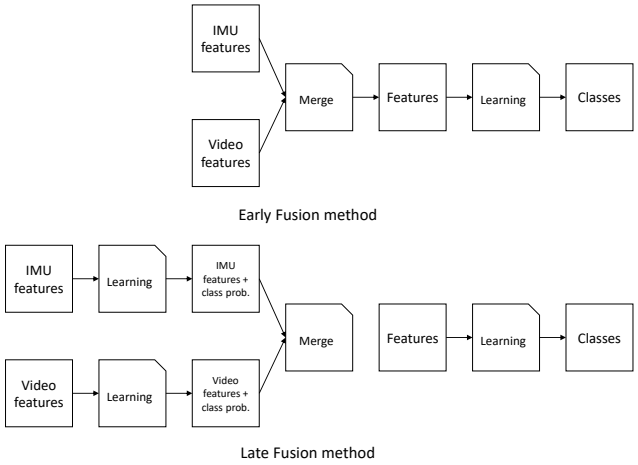


Figure 29: Pipeline for the fusion of the modalities. The top pipeline shows our early fusion method, the bottom one our late fusion approach.

the CMU-MMAC dataset, this step could only be done on our own dataset.

6.4.3 *Combining Both Modalities*

Given the two modalities, we now try to estimate the activities that the subjects perform. For that purpose, we have to define a method to combine both features to be used in one machine learning model. Before we combine the data, we first have to align both modalities which we described in Section 6.3. Since the data may start at different times, we consider the biggest temporal overlap of the data. Here, we consider the latest starting point and the earliest ending point among all modalities for each scenario. Points of data that are earlier or later than the respective starting and endpoints are not considered for the experiments and are discarded. The resulting data will later be used for training and testing. From the trimmed data we calculate our features as described before. As our windows have temporal information, we can map the windows to each other and have one consistent dataset. Consider as an example a video window w_{vid} with start and endpoints t_s and t_e. When windows are of the same length for both inertial and video data, we can map them one to one. Otherwise, we can find all matching inertial windows w_{imu} by filtering for start and end time and requiring them to be in the range of t_s – t_e. The first approach we test is early fusion, where we concatenate the feature vectors of all three modalities and learn one model. This way, all information is immediately present in the model and classifiers can choose what features to use. For the scenarios where the window size varies, we order the matching inertial windows by time and concatenate them in that order. Figure 29 shows a simplified flow diagram of the fusion approach (seen at the top). For simplicity, the inertial data is shown as one flow, though we use two sensors

there. Another approach is late fusion learning, seen at the bottom of Figure 29. Here, we first concatenate both inertial windows and learn a model for this subset of features. Simultaneously, we learn a model for the image windows. For both modalities, we return the class probabilities and append them to the feature vectors. Finally, we concatenate both feature vectors with the added probabilities to one big feature vector. Using these combined features, we once again learn a model to predict the activities. This approach separates the modalities first, thus giving each model the chance to learn specifics for each modality. Furthermore, we can leverage different machine learning algorithms for each sensor type. As the features for each modality represent different aspects of the activity, using separate algorithms could be beneficial for the overall results.

To gain more insights about each modality, we report the performance of each sensor separately in addition to the final performance. This way, we can also see if the different sensors work best with different learning algorithms. The next section describes the experiments in greater detail and presents the results.

6.5 EXPERIMENTS

6.5.1 *ADL Dataset*

For the experiments, we consider each subject separately and test our model with a cross-validation. A cross-subject setting could be used with a bigger dataset, but since this dataset includes two subjects it is not feasible to learn a model this way. To test for stability in the smaller datasets, we run each cross-validation 100 times with different folds and check for similar results. As we want to have a deeper insight into the influence of each modality in combination with different classifiers, we test different combinations of classifiers for the multimodal settings. Configuration parameters include the classifier that is used for the late fusion learning, which modalities are used, and whether ground truth or the neural network bounding boxes are used. For classification, we use Random Forest and Logistic Regression algorithms. We also tested other classifiers like SVM, but the results were most promising with the algorithms mentioned above. When we consider all modalities, the classifiers used for the separate sensors are Random Forest for acceleration data and Logistic Regression for vision data. This way we keep the single modalities fixed and only change the fusion learning algorithm, reporting its performance at the end. We also tested early fusion, but this yielded an overall performance loss for the classification in our cross-validation evaluation.

Using a sliding window approach with overlap poses a problem: two consecutive windows may end up in the training and the testing set respectively. Since windows are overlapping, theoretically these overlapping parts of the data are part of windows in both training and testing. To avoid this, we sampled our data depending on which

modalities we evaluate, making sure that no data is present in training and testing simultaneously. In both the vision and combined approach, the windows are based on the vision windows. As it has an overlap of 50%, we consider every other window and, in the case of the combined approach, the respective IMU window. When considering only acceleration data, the overlap of windows is 75%, thus we consider every fourth data point in the experiments. In this specific case, the amount of data available can be fairly small for some of the very short activities, resulting in folds with very few instances for some classes. Therefore, we use a five-fold validation in these scenarios instead of a 10-fold cross-validation. The results are reported as an average of both test subjects.

Config	**Precision**	**Recall**	**F_1-score**
RF_IMU	0.673	0.556	0.609
LR_IMU	0.516	0.392	0.446
RF_VIS_GT	0.872	0.622	0.726
LR_VIS_GT	0.855	0.590	0.698
RF_VIS_LEARN	0.506	0.367	0.425
LR_VIS_LEARN	0.721	0.337	0.460
RF_ALL_GT	0.843	0.754	0.796
LR_ALL_GT	0.897	0.753	0.819
RF_ALL_LEARN	0.816	0.709	0.758
LR_ALL_LEARN	0.880	0.722	**0.793**

Table 14: Different configurations for our learning method. Values are reported as an average over each class and for both subjects. RF = Random Forest, LR = Logistic Regression, ALL = both modalities were used, VIS = only vision features, IMU = only acceleration features, GT = ground truth vision, LEARN = vision features that have been detected by our neural network.

Table 14 shows that the best configuration uses all modalities and Logistic Regression as the fusion learning algorithm, yielding a F_1-measure of 79.3% (leaving out the ground truth vision scenarios). As expected, the results for using only vision features are far higher when assuming perfect vision. The gap in performance can most likely be attributed to the object detection algorithms that we use. Especially with our scenarios including different environments and a camera sensor of lower quality, pre-trained object detection can still classify many objects wrongly. With a bigger dataset, a custom model could be trained that may yield an improvement for the vision results. Considering the results of the inertial data classification, a great difference in performance among the learning algorithms is visible. This is in line with the analysis of related work that shows that Random Forest classification works well with inertial windows [87]. The combination of the sensors, however, is helping the results overall. Especially given the fact that these results are achieved with a pre-trained object detection model. Overall, the results of the classification tend

to prefer a high precision at the cost of recall which is beneficial in our scenario, as a sequence of correctly classified activities with some windows not assigned at all can still be used to reconstruct the correct order of activities. In the next step, we take a closer look at the separate classes and their performance using the best configuration from the previous experiment.

Class	**Precision**	**Recall**	**F_1-score**
none	0.928	0.986	0.956
drink_water	0.886	0.62	0.729
eat_banana	0.868	0.511	0.643
eat_bread	0.867	0.749	0.804
prepare_bread	0.891	0.929	0.909
take_meds	0.894	0.676	0.769
wipe_mouth	0.837	0.585	0.688

Table 15: A closer look at the results for our best configuration for each activity separately. Both vision and acceleration features are used in combination with Logistic Regression.

Table 15 shows the results for all classes, broken down for each class separately. At first glance it can be seen that the performance varies among the different activities. A great performance can be achieved for the bread preparation class, with an F_1-measure of 90.9%. One possible explanation for the good performance is the uniqueness of the features for both modalities we use. In the case of inertial data, the motion of buttering a piece of bread is distinctively different from the other activities which all involve some sort of grabbing or lifting motion. For the video data, this scenario also offers unique views, as the test subjects were looking down on their plate and focusing on it and the bread. Most of the other classes are performed with an overlook of the table, thus resulting in a similar scenery. Additionally, this activity, in combination with eating bread, was performed the longest by the subjects, yielding more instances for training.

Eating a piece of banana and wiping the mouth after eating are the worst-performing activities, yielding F_1-measures of 64.3% and 68.8% respectively. There are separate reasons for both classes. In the case of eating a piece of banana, the shortness of the activity is the main problem. Test subjects were eating just one piece of fruit which is readily available on the table. Thus, the activity is very short, only offering few unique aspects to be learned. Wiping the mouth has the issue of hard to detect objects. The napkin is often only partially visible, parts of it hidden underneath a plate. This makes it difficult for the object detection algorithm to detect the object.

6.5.2 *CMU-MMAC Dataset*

For the experiment on the CMU-MMAC dataset with the original annotations, we evaluate the whole dataset among all subjects. Another

approach would be to learn classifiers for each subject individually, but this is not feasible with the amount of data available. Here we also run the experiment 100 times and calculate the average precision, recall, and the resulting average F_1-measure.

Config	Precision	Recall	F_1-score
RF_ALL	0.748	0.436	0.551
LR_ALL	0.738	0.482	**0.584**
RF_IMU	0.727	0.440	0.548
LR_IMU	0.230	0.115	0.153
RF_VIS	0.400	0.269	0.321
LR_VIS	0.395	0.236	0.295

Table 16: Results for CMU-MMAC dataset. Here we use the same method as above for our experiment. As we do not have bounding-box ground truth data, we can only learn on the output of our neural network.

Given the harder task of the CMU-MMAC dataset, we achieve a lower F_1-score of 58.4% (see Table 16). This is not surprising, given the setting of the dataset, where a larger amount of subjects perform a greater set of activities, both adding more variation to the dataset. The bad performance using the vision features is also striking, with the performance going down to 32.1%. One explanation for this score is the reduction of the annotations to just the verb part. Annotations for the dataset are provided in the form of *verb-object1-preposition-object2*. As this results in a very huge set of labels with small amounts of instances per label, we reduce the annotations to just the verb. Thus, activities like *open-brownie_box* and *open-cupboard_top_left* are assigned the same label, even though they are performed on very different objects and in different situations. Vision features in this context are relying on the objects visible in the frame and thus have issues to properly differentiate the various activities. What is also striking is the fact that on this dataset the vision features perform better with a Random Forest instead of the Logistic Regression as it was the case in the ADL dataset. When looking at the acceleration data though, the results are fairly good. This is in line with results in [31] (Chapter 3) where it was shown that hierarchical clustering of the activities tends to favor activities with the same verb. Therefore, acceleration data is able to represent similar activities in a similar fashion. However, in the context of this dataset, Logistic Regression does not seem to be able to properly learn a model for the inertial data. We could already see that Logistic Regression performs worse on our dataset when applied to acceleration data. This effect is even stronger in the CMU-MMAC dataset, most likely because of the bigger set of labels that have to be recognized. Random Forest behaves similar in both cases and yields good results which is in line with previous research [87]. As object annotations for the videos are not

present in the CMU-MMAC dataset, we cannot run experiments on perfect vision.

To look deeper into the classification results, we consider the IMU classification results on their own and show the performance for each class.

Class	Precision	Recall	F_1-score
close	0.516	0.062	0.111
crack	0.757	0.389	0.514
none	0.674	0.783	0.724
open	0.690	0.481	0.567
pour	0.601	0.613	0.607
put	0.752	0.460	0.571
read	0.834	0.551	0.664
spray	0.890	0.726	0.800
stir	0.744	0.811	0.776
switch_on	0.859	0.630	0.727
take	0.708	0.648	0.677
twist_off	0.824	0.188	0.306
twist_on	0.793	0.196	0.314
walk	0.695	0.215	0.328

Table 17: A closer look at our best performing configuration for the classes in the CMU-MMAC dataset. The model was learned in a 10-fold cross-validation among all subjects.

Table 17 shows our findings. Good performance can be seen in classes like spraying and stirring with a F_1-score of 80% and 77% respectively, while generic classes like walking or closing are not recognized very well. This seems to be in line with our assumption that the acceleration data is able to distinguish specific activities (i.e. stirring involves a motion that is very unusual compared to the others) and has problems distinguishing verbs that are very generic.

To compare our results, we evaluate against a previous approach [102] that uses the same scenario for their dataset (i.e. the brownie recipe of the CMU-MMAC dataset) and also the same approach for reducing the labels. They use a novel classification approach on SIFT features from the video frames of the dataset. To fit the evaluation of the work, we modify our training to use the first eight of the test subjects for training and the last four for testing. In this scenario, we also used resampling of the data to simulate an even class distribution. We report the results in the form of the F_1-measure for each class. It can be seen that with the exception of the pour and the none class, our approach outperforms previous results. Overall, this evaluation setting shows a performance drop, as we consider a fixed split that only allows for a small training set. This way, we are also encountering the difficult problem of cross-subject learning, which we did not consider in the previous experiments. What can be seen though, is that some classes

Class	Baseline*	SSVM*	PR-SSVM*	Our approach
close	0.0	0.006	0.01	**0.045**
crack	0.065	0.035	0.053	**0.124**
none	0.075	0.195	**0.251**	0.198
open	0.098	0.124	0.152	**0.181**
pour	0.140	0.266	**0.276**	0.126
put	0.087	0.079	0.121	**0.247**
read	0.0	0.008	0.037	**0.039**
spray	0.016	0.013	0.016	**0.074**
stir	0.352	0.148	0.294	**0.587**
switch_on	0.038	0.043	0.042	**0.098**
take	0.075	0.195	0.139	**0.234**
twist_off	0.0	0.024	0.036	**0.055**
twist_on	0.0	0.02	0.025	**0.047**
walk	0.0	0.0	0.083	**0.094**

Table 18: Comparison against state-of-the-art approach. Values marked with a * are directly taken from [102]. Here the model is learned on 8 subjects and tested on the remaining 4.

like stirring, putting and taking can be learned across subjects given enough training-data. Evidently, these are also among the classes that occurred the most in the dataset (see Section 6.3, Figure 25).

We can see that the combination of inertial and video data yields a better result than each sensor on its own. Depending on the activity that should be recognized, modalities perform differently as they are relying on the variation within the data. Inertial data, for example, may not be as expressive when the activities that are to be distinguished are very similar in motion. Thus, it makes sense to consider the combination of both modalities to predict high-level activities.

6.5.3 *CMU-MMAC - New Annotations*

Next, we consider the new annotations provided by [101] to learn on an even bigger set of activities for the CMU-MMAC dataset. As done with the original annotations, we test early and late fusion approaches in this scenario. For our experiments, we consider the annotations for the Brownie scenario with 28 different test subjects. With the increased dataset, however, it was also feasible to run a grid-search on the dataset to properly tune the classifier. Here, we use a fixed split for training and test data, with a split of 80% for training and 20% for testing. Then we run a grid-search with a 5 fold cross-validation on the training data for each classifier, finally evaluating on the test dataset. For the random forest we tune these parameters:

- Number of estimators
- Maximal depth of trees

- Min samples per leaf and per split
- The number of features to consider when splitting (all, or $\sqrt{n\_features}$)

For the logistic regression we consider:

- Number of iterations
- Optimizer type (newton, simple)
- Distance C

Config	Precision	Recall	F_1-score
LR_EARLY	0.430	0.326	0.337
LR_LATE	0.378	0.323	0.329
RF_EARLY	0.831	0.604	**0.664**
RF_LATE	0.572	0.626	0.574

Table 19: Overall performance of different classifiers using early and late fusion. Late fusion dropped in this scenario.

Results on the new dataset improve, with the new best model improving the F_1-measure by 8%. After running all experiments, we can see that the performance for logistic regression is worse than the random forest. These results differ from the previous experiments. It suggests, that the logistic regression cannot fully abstract on a bigger dataset and thus the random forest is the overall better choice. Fully comparing the results is difficult, however, as the annotations for the dataset are similar to the original, but not the same. For the next step, we again look at the performance of the single classes to see if similar patterns can be seen.

Class	Precision	Recall	F_1-score
clean	0.947	0.409	0.571
close	0.764	0.479	0.589
fill	0.748	0.967	0.844
open	0.720	0.696	0.707
other	0.866	0.615	0.719
put	0.665	0.537	0.595
shake	1.000	0.270	0.426
stir	0.904	0.977	0.939
take	0.620	0.595	0.607
turn_on	0.976	0.840	0.903
walk	0.935	0.256	0.402

Table 20: Detailed evaluation for classes in the bigger CMU subset with the best configuration which is a random forest with early fusion.

Table 20 shows the results of our run on the greater subset of the CMU dataset for each activity separately. Overall the results are very

promising and show an improvement to the previous experiments. This makes sense, as the dataset size is increased greatly. Some trends that could be seen in the previous experiments are also present in the results of this experiment. Stirring, a class with a very unique motion and long sequences of data, can be recognized fairly well. Classes like walking though, are still hard to classify as they do not contain enough inertial cues in the wrist-worn sensor. Considering a sensor that is attached to the legs may yield better results, but would increase the overall amount of sensors which is why we left it out. A direct comparison to the original data is difficult though, as the annotations are not done by the same annotators and also use different classes. It can be seen though, that even the classes that were very difficult to classify with the original annotations (e.g. walking and closing) have improved with the bigger dataset. Overall, using a fused approach with a multimodal setting seems to be promising to classify human activities.

6.6 DISCUSSION

The results of the experiments show, that combining vision and inertial data is a promising approach for classifying human activities. It is helpful especially in those cases, where either of the modalities is not capable of capturing specific aspects of an activity. An example would be the consumption of a snack compared to the intake of medicine where an inertial sensor may have problems distinguishing the activity, as it relies to some extent on the objects used. However, the approach can still be extended. Estimating interactions with objects is one important aspect. Using the overlap of a hand with an object can yield good results, but especially in frames with many objects, a lot of overlap can exist. In these cases, motion tracking information of objects could help, as it may be used to detect the movement of objects. However, motion tracking is especially difficult in a scenario, where an egomotion of the camera is present which is the case in the datasets we consider. Furthermore, as there is no depth information present in the data, an overlap cannot fully represent the interaction.

Another aspect to consider in this work is the issue of privacy. Systems that recognize activities always bear the challenge of privacy concerns, especially when video cameras are used in the process. When a video camera is recording a user or from a user's perspective for a long period of time, it may capture activities that are deemed sensitive. We believe that smart devices can help to mitigate the privacy concerns that arise when using cameras. For one, processing and calculation of the data can be done offline within the home environment where such a system is in use. Additionally, when on-the-fly classification becomes feasible, video data may not even be stored but just processed as a stream, in the end only using object information. In this context, if the set of objects that can be detected is kept to a minimum, the amount of sensitive information processed can be reduced greatly. This way, potentially no data is leaked to the outside

which may help to mitigate possible concerns. Another aspect could be the utilization of the smart device to recognize the context of a user. The camera could for example be turned off when a user is in a certain room or at a certain time where their privacy concerns are very strong (e.g. in the context of personal hygiene).

We also considered using other sensors for the recognition, like depth or infrared cameras that may seem less intrusive at a first glance. The obvious downside of these devices is their relative low availability in smart devices, making it difficult to easily use them with current technology. Depth cameras, for instance, have become more common in consumer hardware in recent years, but still are not as prevalent as standard cameras. Additionally, the amount of sensitive information collected by these types of cameras is comparable to that of a standard camera and in some cases even higher, making the privacy concerns an even harder problem to solve. Infrared cameras for instance can relay much more information about a person that is recorded just by the temperature data it collects. On top of that, there are also practical issues in our scenario. Depth cameras, for example, are bound to a minimum and maximum distance they are able to capture. With a person wearing such a camera for an egomotion recording, many interactions close to the user may not be captured by the camera. Overall, the usage of cameras can be challenging in a live system, but we believe that considering the added information gain of the modalities and using a proper and privacy-aware implementation, such challenges may be overcome.

6.7 CONCLUSION AND FUTURE WORK

In this work, we presented a new multimodal dataset that includes activities of daily living. It poses the challenge of similar activities, namely food and water consumption, and medicine intake. All activities in the dataset were performed by two subjects at two different locations. The collected data includes acceleration, gyration and magnetic field data from 6 different body positions and videos from three different angles, two of which are egocentric. Based on this dataset we present a method for recognizing activities, using window features with fused video and acceleration data. Here, we use time and frequency domain features for the acceleration data and object information, encoding hand interactions, for the vision data. For the recognition of objects in a frame, we utilize a pre-trained neural network where we use the overlap of the subject's hand with objects in a frame as a feature. After learning a model for each modality separately, we fuse them together and learn an overall model using Random Forest and Logistic Regression classifiers. This way, we were able to achieve an F_1 measure of 79.6% on our presented dataset and 58.4% on the CMU-MMAC dataset (66.4% for the bigger subset). We also show that we beat a state of the art activity recognition approach for the CMU-MMAC dataset. Both scenarios (ADL and cooking) pose different challenges for our approach. For our dataset, the similarity

of the activities is challenging when considering acceleration data, as the difference of the actions is mostly rooted in the interaction with different objects. The CMU-MMAC dataset contains a wider variety of activities by a greater number of subjects, thus including more variation in the data. We can show that our approach is promising for the recognition of activities in a multimodal setting, including the usage of off-the-shelf sensors build into smart devices. Especially when utilizing the new bigger set of annotations for the CMU-MMAC dataset, we could see that results improve when more data is available.
In future work, some aspects of the method could be adapted. For the features, parts like the object detection network could be exchanged. If we are able to get bigger sets of object annotations also for the CMU dataset, transfer-learning a model may be a feasible approach to get better object information. We could also re-evaluate the selection of modalities. So far, we focus on a relatively small subset of modalities to analyze, as we want to utilize as few devices as possible. Still, it may be interesting to evaluate different and greater sets of modalities for our goal, keeping in mind not to over-fit the approach. Gyration and magnetic field data are obvious candidates, as they are recorded alongside the same sensors already. Fusion techniques could also be changed, where for example different lengths of windows are used for each modality. Between this bigger set of overlapping windows, boosting or voting mechanism could learn the best fusion strategy.

Part III

OUTLOOK

7 CONCLUSION

In this work, we analyze the process of activity recognition from the labeling effort up to the classification task. We could see that activity recognition has multiple applications in the real world, making it an interesting and viable topic of research, even for the future. Specifically, the multimodality of our approaches yields interesting results and a contribution to the research field.

Compared to many other fields, data gathering and annotation take a significant amount of time for relative small dataset sizes. There are multiple reasons for that. Activities can be partially occluded or overlapping which makes identifying the start and the end of the specific activity difficult. Additionally, since we are working in a multimodal setting, annotations are typically made on one modality and then mapped to the others. As this is often done on video data, decisions about the start and the end of single activities can vary greatly among different annotators, especially when the activities are strongly interleaved. On top of that, the granularity of the target-activities can have another huge impact on the annotation time. Here, annotating activities top down (i.e. first activities, then actions) can speed up the process and make the annotations more consistent. We could show that using templates of activities, we can make labeling suggestions that are very close to the actual activity. This was evaluated by measuring the temporal distance of the suggestion to an actual activity in our datasets. The method was evaluated on two different datasets, both dealing with arm movement activities. Overall, a need for more intuitive and fast annotation tools is still present in the field. Thus more research can be conducted here.

We then looked at different methods for the segmentation of raw sensor data. Activity recognition often includes the detection of activities of various lengths. Therefore, we wanted to investigate if it is possible to separate them before we classify the segments. For this purpose, we tested a set of methods to see how well we can split up the raw data only based on inertial information. We could see, that the results are not satisfactory with the methods we evaluated. This is due to the fact that the dataset we considered has a lot of interleaved activities as well as ambiguous motions with different labels. Therefore, typical indicators like energy values going up due to the start of a movement may not work properly in this scenario. Our work shows that activities in scenarios with more complexity are way harder to segment when compared with typical locomotive activities.

For the final part of our work, we examined two use cases for activity recognition and evaluated how multimodal approaches influence

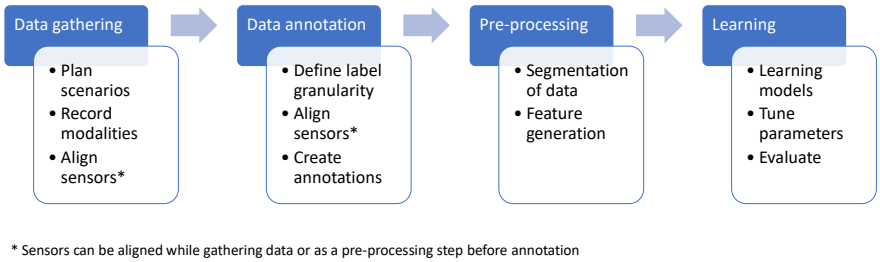


Figure 30: Typical pipeline for activity recognition

the results. When we consider the industry case in the form of our picking dataset, we could see promising results in the form of detecting picking motions. Here, the multimodal aspect of our approach could be helpful, as results were improving greatly when the combination of sensors was used. With the addition of barcode scanning, such systems could run in the near future, making hands-free operations more feasible in warehouses. In the second case, we looked at different fusion and classification methods for multimodal activity recognition in different scenarios. Namely cooking activities and activities of daily living. For these scenarios, we looked at early and late fusion methods to classify movements of test subjects and also compared the impact of the different modalities on the classification results. It could be seen that the combination of the modalities works well together, giving the overall best results. Vision on its own is giving great precision while inertial data yields better recall. Our vision features based on pre-trained neural networks for object detection could improve the results compared to earlier work in the field. In so far, our experiments were indicating a promising trend for multimodal activity recognition.

If we consider Figure 30, we can see that our work touched upon every aspect of the activity recognition pipeline. We have provided a new dataset for ADL activities that are performed in different human body poses, with a focus on hard to distinguish target classes involving the consumption of food and medicine. Then, based on data for warehouse picking and our ADL dataset, we developed a method for labeling support that allows us to speed up the process. Segmentation was explored by us, though we could not find a reasonable method for our use cases as we consider fairly complex activities. Finally, in our work, we have contributed to the field of activity recognition with multiple publications that explore new methods for feature generation in both industry and personal settings. In general, we can see that the whole field of activity recognition offers a lot of directions researchers can examine, with many sub-tasks being available for exploration. We analyzed a small part of multimodal activity recognition, leaving out bigger fields like event-based approaches and multi-user scenarios among other topics. As the trend for more smart devices and also smart sensors continues (as can be seen with the trend in IoT), new possibilities for better recognition systems open up. Up to this point one of the main applications of activity recogni-

tion in the personal sector is for sports activities and personal health as well as navigation to some extend. However, activity recognition is currently lacking when it comes to the interaction of a person with their environment in the digital space. Some environments start to be digitized, but the different devices are usually considered independently. In this sense, activity recognition could benefit from approaches that are present in the field of augmented reality. For more sophisticated ubiquitous computing solutions, the interoperability of different devices has to be ensured. Ideally, it would be possible to have ad-hoc systems that can adapt easily to the environment that they are used in.

8

FUTURE WORK

While we did work on the full pipeline of activity recognition, many things are left to be explored in this field of research that we could not consider at this point in time. We take a look at each subtask of the pipeline to identify possible extensions and new directions research can take to extend the field. In the first step, we have to consider data collection. Right now the community does not have a lot of adequately sized datasets to develop and evaluate their methods, as collecting them for multimodal scenarios is very time-consuming. Errors while recording are often difficult to fix and with an increased number of sensors, the amount of points of failures grows as well. What we do find, however, are many vision-based datasets that unfortunately do not contain inertial data information since the focus is mostly on purely vision-based solutions. Here, a huge effort has to be undertaken to close the gap and make more sophisticated approaches viable. An idea could be to crowdsource the collection of the data. In recent years, the amount of people streaming their lives online has increased significantly, with many people streaming parts of their everyday life like cooking, playing music, or other hobbies. Researchers could try to build tools that make it easy for volunteers to collect data during their stream, e.g. by collecting video feeds in addition to smartwatches worn by the streamer.

For the step of data annotation, the main focus should be ease of annotation. Assuming larger datasets can be gathered sufficiently, the labeling of such datasets should be made easier to accommodate for the amount of data. With our contribution, we showed a method for faster labeling tools by utilizing templates for actions based on inertial data. What was left unexplored is the usage of video data for faster annotation. If labels are backed by some form of knowledge, we may be able to associate objects with activities. By detecting said objects, the task of finding activities could be sped up even more. Combining this approach with a well-made crowdsourcing solution (similar to the implementation in [90]) would enable the community to build datasets more easily.

The next step in the pipeline is the segmentation of the data. Our analysis in this part was rather rudimentary and many more approaches could be tested. As already mentioned in the labeling step, considering the video information could be a big part of future work. While it is true that a limited field of view may not capture the exact change of activity, cameras are constantly improving, with better viewing angles and higher resolutions becoming more common over time. This, in addition to background knowledge about activities and objects, could make a segmentation based on video data possible and maybe

even lead to more unified approaches of segmentation and classification of activities.

Lastly, we consider the feature engineering and the classification of the activities. One direction could be the usage of elaborate deep learning approaches. It has been shown in recent years that neural networks can cope well with multimodal input data, even from different types of sensors [71]. The advantage here would be that less or even no manual feature generation has to be done, as the network explores the data on its own, finding correlations between different sensors independently and even creating its own representations. Such a unified approach would have to be compared to traditional sensor fusion methods to see how much improvement can be made. In other fields, like computer vision, it already has been shown that an end to end approach using neural networks matches and outperforms manual feature generation. It would be interesting to see how well this phenomenon translates to activity recognition.

Furthermore, exploring the usage of transfer learning to make modern neural network methods easily applicable for activity recognition would be in a similar category. Here, researchers would have to first identify which networks contain sufficiently similar knowledge such that it is feasible to transfer it to the activity recognition scenario. One such candidate could be the active research field of object detecting neural networks. It would be interesting to see, if an object detecting network could easily be adapted to an activity recognition network. Potential challenges are maybe posed by activities that share a similar view in single frames but that involve very different movements. Here, the addition of inertial data to the network input could be beneficial. Considering short video sequences as an input instead of single frames could be another approach. At this point, it still has to be shown that the transfer of a pre-trained model with so many alterations to the network is yielding positive results.

Alternatively, a reasoning based approach could be another viable solution for a more complex activity recognition system. Assuming that single parts of activities can be detected reliably, the results of those models may be collected into one cohesive sequence of activities and/or actions. Using some sort of ontology that captures the activities that are to be recognized, reasoning solutions may construct higher levels of activities they derived from the more fine-grained input sequences. In this case, it would be especially interesting to analyze how fine-grained the target-classes of the initial models are set and respectively how much engineering effort has to be put into designing the ontology. If we assume that different modalities are best captured by different machine learning approaches (which our work in Chapter 6 indicates), then reasoning based solutions gain a significant advantage as they are not bound to have one unified model for all modalities. They could also be used to extend existing smart home activity recognition systems. On top of adding new activities that the

system recognizes, existing detections may be improved with the new input source of personal sensors.

With the increasing digitization of everyday life, activity recognition starts playing a bigger role in many peoples lives. In addition to more methods and algorithms being proposed by researchers, we can also see other areas of research influencing activity recognition and even touching upon similar topics. Virtual and augmented reality often deal with challenges similar to personal sensor-based activity recognition, while the trend of IoT could be especially useful in event-based activity recognition. These trends show that some of the ideas of ubiquitous computing are getting realized in recent years, opening up new challenges for the research community. Finally, the impact of such systems on a user's everyday life should also be considered. Concerns about privacy and strong reliance on systems (potentially to the extent of creating a single point of failure) are just a few examples of activity recognition having an impact on everyday life. Thus, studies ideally should always keep these impacts in mind.

BIBLIOGRAPHY

[1] Girmaw Abebe and Andrea Cavallaro. "Hierarchical modeling for first-person vision activity recognition." In: *Neurocomputing* 267.Supplement C (2017), pp. 362–377. DOI: 10.1016/j.neucom.2017.06.015.

[2] Valentina Agostini, Gabriella Balestra, and Marco Knaflitz. "Segmentation and classification of gait cycles." In: *IEEE Transactions on Neural Systems and Rehabilitation Engineering* 22.5 (2013), pp. 946–952.

[3] Valentina Agostini, Laura Gastaldi, Valeria Rosso, Marco Knaflitz, and Shigeru Tadano. "A wearable magneto-inertial system for gait analysis (H-Gait): Validation on normal weight and overweight/obese young healthy adults." In: *Sensors* 17.10 (2017), p. 2406.

[4] SJ Allin, A Bharucha, John Zimmerman, D Wilson, MJ Robinson, Scott Stevens, H Wactlar, and CG Atkeson. "Toward the automatic assessment of behavioral disturbances of dementia." In: *Ubiquitous Computing (UbiComp), 2003 International Conference on*. 2003.

[5] Oliver Amft and Kristof Van Laerhoven. "What will we wear after smartphones?" In: *IEEE Pervasive Computing* 16.4 (2017), pp. 80–85.

[6] Leopoldo Armesto, Stefan Chroust, Markus Vincze, and Josep Tornero. "Multi-rate fusion with vision and inertial sensors." In: *Robotics and Automation, 2004. Proceedings. ICRA'04. 2004 IEEE International Conference on*. Vol. 1. IEEE. 2004, pp. 193–199.

[7] Ferhat Attal, Samer Mohammed, Mariam Dedabrishvili, Faicel Chamroukhi, Latifa Oukhellou, and Yacine Amirat. "Physical human activity recognition using wearable sensors." In: *Sensors* 15.12 (2015), pp. 31314–31338.

[8] Ihn-Han Bae. "An ontology-based approach to ADL recognition in smart homes." In: *Future Generation Computer Systems* 33 (2014), pp. 32–41.

[9] Jens Barth, Cäcilia Oberndorfer, Cristian Pasluosta, Samuel Schülein, Heiko Gassner, Samuel Reinfelder, Patrick Kugler, Dominik Schuldhaus, Jürgen Winkler, Jochen Klucken, et al. "Stride segmentation during free walk movements using multi-dimensional subsequence dynamic time warping on inertial sensor data." In: *Sensors* 15.3 (2015), pp. 6419–6440.

[10] Michael Barz, Mohammad Mehdi Moniri, Markus Weber, and Daniel Sonntag. "Multimodal Multisensor Activity Annotation Tool." In: *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct*. New York, NY, USA: ACM, 2016, pp. 17–20. DOI: 10.1145/2968219.2971459.

[11] Anabela Berenguer, Jorge Goncalves, Simo Hosio, Denzil Ferreira, Theodoros Anagnostopoulos, and Vassilis Kostakos. "Are Smartphones Ubiquitous?: An in-depth survey of smartphone adoption by seniors." In: *IEEE Consumer Electronics Magazine* 6.1 (2016), pp. 104–110.

[12] Donald J. Berndt and James Clifford. "Using Dynamic Time Warping to Find Patterns in Time Series." In: *Proceedings of the 3rd International Conference on Knowledge Discovery and Data Mining*. AAAI Press, 1994, pp. 359–370.

[13] A. Betancourt, P. Morerio, C. S. Regazzoni, and M. Rauterberg. "The Evolution of First Person Vision Methods: A Survey." In: *IEEE Transactions on Circuits and Systems for Video Technology* 25.5 (2015), pp. 744–760. DOI: 10.1109/TCSVT.2015.2409731.

[14] Axel H Boersch-Supan. "Aging in Germany and the United States: international comparisons." In: *Studies in the Economics of Aging*. University of Chicago Press, 1994, pp. 291–330.

[15] G. Bradski. "The OpenCV Library." In: *Dr. Dobb's Journal of Software Tools* (2000).

[16] Andreas Bulling, Ulf Blanke, and Bernt Schiele. "A tutorial on human activity recognition using body-worn inertial sensors." In: *ACM Computing Surveys (CSUR)* 46.3 (2014), pp. 1–33.

[17] Sait Celebi, Ali Selman Aydin, Talha Tarik Temiz, and Tarik Arici. "Gesture recognition using skeleton data with weighted dynamic time warping." In: *VISAPP (1)*. 2013, pp. 620–625.

[18] Ricardo Chavarriaga, Hesam Sagha, Alberto Calatroni, Sundara Tejaswi Digumarti, Gerhard Tröster, José del R Millán, and Daniel Roggen. "The Opportunity challenge: A benchmark database for on-body sensor-based activity recognition." In: *Pattern Recognition Letters* 34.15 (2013), pp. 2033–2042.

[19] Chen Chen, Roozbeh Jafari, and Nasser Kehtarnavaz. "Utd-mhad: A multimodal dataset for human action recognition utilizing a depth camera and a wearable inertial sensor." In: *Image Processing (ICIP), 2015 IEEE International Conference on*. IEEE. 2015, pp. 168–172.

[20] Tiziana D'Orazio, Marco Leo, Nicola Mosca, Paolo Spagnolo, and Pier Luigi Mazzeo. "A semi-automatic system for ground truth generation of soccer video sequences." In: *Sixth IEEE International Conference on Advanced Video and Signal Based Surveillance*. IEEE Computer Society, 2009, pp. 559–564.

[21] Navneet Dalal and Bill Triggs. "Histograms of oriented gradients for human detection." In: *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*. Vol. 1. IEEE. 2005, pp. 886–893.

[22] Dima Damen et al. "Scaling Egocentric Vision: The EPIC-KITCHENS Dataset." In: *European Conference on Computer Vision (ECCV)*. 2018.

[23] René De Koster, Tho Le-Duc, and Kees Jan Roodbergen. "Design and control of warehouse order picking: A literature review." In: *European journal of operational research* 182.2 (2007), pp. 481–501.

[24] Fernando De la Torre Frade, Jessica K Hodgins, Adam W Bargteil, Xavier Martin Artal, Justin C Macey, Alexandre Collado I Castells, and Josep Beltran. *Guide to the Carnegie Mellon University Multimodal Activity (CMU-MMAC) Database*. Tech. rep. CMU-RI-TR-08-22. Pittsburgh, PA: Robotics Institute, Apr. 2008.

[25] Debraj De, Pratool Bharti, Sajal K Das, and Sriram Chellappan. "Multimodal wearable sensing for fine-grained activity recognition in healthcare." In: *IEEE Internet Computing* 19.5 (2015), pp. 26–35.

[26] Manfred Del Fabro and Laszlo Böszörmenyi. "AAU Video browser: Non-sequential hierarchical video browsing without content analysis." In: *International Conference on Multimedia Modeling*. Springer. 2012, pp. 639–641.

[27] Manfred Del Fabro, Bernd Münzer, and Laszlo Böszörmenyi. "Smart video browsing with augmented navigation bars." In: *International Conference on Multimedia Modeling*. Springer. 2013, pp. 88–98.

[28] Yueng Santiago Delahoz and Miguel Angel Labrador. "Survey on fall detection and fall prevention using wearable and external sensors." In: *Sensors* 14.10 (2014), pp. 19806–19842.

[29] Alexander Diete and Heiner Stuckenschmidt. "Fusing object information and inertial data for activity recognition." In: *Sensors* 19.19 (2019), p. 4119.

[30] Alexander Diete, Timo Sztyler, and Heiner Stuckenschmidt. "A smart data annotation tool for multi-sensor activity recognition." In: *2017 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops)*. IEEE Computer Society, 2017, pp. 111–116.

[31] Alexander Diete, Timo Sztyler, and Heiner Stuckenschmidt. "Exploring Semi-Supervised Methods for Labeling Support in Multimodal Datasets." In: *Sensors* 18.8 (2018). ISSN: 1424-8220. DOI: 10.3390/s18082639. URL: http://www.mdpi.com/1424-8220/18/8/2639.

[32] Alexander Diete, Timo Sztyler, and Heiner Stuckenschmidt. "Vision and acceleration modalities: Partners for recognizing complex activities." In: *2019 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops)*. IEEE. 2019, pp. 101–106.

[33] Alexander Diete, Timo Sztyler, Lydia Weiland, and Heiner Stuckenschmidt. "Recognizing grabbing actions from inertial and video sensor data in a warehouse scenario." In: *Procedia computer science* 110 (2017), pp. 16–23.

[34] Alexander Diete, Timo Sztyler, Lydia Weiland, and Heiner Stuckenschmidt. "Improving motion-based activity recognition with ego-centric vision." In: *2018 IEEE International Conference on Pervasive Computing and Communications : PerCom 2018, Athens, Greece, March 19-23, 2018 : PerCom Workshops proceedings*. IEEE Computer Society, 2018.

[35] Muhammad Ehatisham-Ul-Haq, Ali Javed, Muhammad Awais Azam, Hafiz MA Malik, Aun Irtaza, Ik Hyun Lee, and Muhammad Tariq Mahmood. "Robust human activity recognition using multimodal feature-level fusion." In: *IEEE Access* 7 (2019), pp. 60736–60751.

[36] Olivier Friard and Marco Gamba. "BORIS: A free, versatile open-source event-logging software for video/audio coding and live observations." In: *Methods in Ecology and Evolution* 7.11 (2016), pp. 1325–1330.

[37] Markus Funk, Alireza Sahami Shirazi, Sven Mayer, Lars Lischke, and Albrecht Schmidt. "Pick from here!: an interactive mobile cart using in-situ projection for order picking." In: *Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing*. ACM. 2015, pp. 601–609.

[38] Eric Guenterberg, Sarah Ostadabbas, Hassan Ghasemzadeh, and Roozbeh Jafari. "An automatic segmentation technique in body sensor networks based on signal energy." In: *Proceedings of the Fourth International Conference on Body Area Networks*. 2009, pp. 1–7.

[39] Anhong Guo, Shashank Raghu, Xuwen Xie, Saad Ismail, Xiaohui Luo, Joseph Simoneau, Scott Gilliland, Hannes Baumann, Caleb Southern, and Thad Starner. "A comparison of order picking assisted by head-up display (HUD), cart-mounted display (CMD), light, and paper pick list." In: *Proceedings of the 2014 ACM International Symposium on Wearable Computers*. 2014, pp. 71–78.

[40] Haodong Guo, Ling Chen, Liangying Peng, and Gencai Chen. "Wearable sensor based multimodal human activity recognition exploiting the diversity of classifier ensemble." In: *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing*. 2016, pp. 1112–1123.

[41] Toshio Hori, Yoshifumi Nishida, and Shin'ichi Murakami. "Pervasive sensor system for evidence-based nursing care support." In: *Robotics and Automation, 2006. ICRA 2006. Proceedings 2006 IEEE International Conference on*. IEEE. 2006, pp. 1680–1685.

[42] Ling-feng Hsieh and Lihui Tsai. "The optimum design of a warehouse system on order picking efficiency." In: *The International Journal of Advanced Manufacturing Technology* 28.5-6 (2006), pp. 626–637.

[43] J. Huang et al. "Speed/accuracy trade-offs for modern convolutional object detectors." In: *Computer Vision and Pattern Recognition (arXiv.org)*. Computer Research Repository, 2016, pp. 1–21.

[44] Amazon Technologies Inc. "Transitioning items from a materials handling facility." Patent US20150012396A1 (US). Jan. 2015.

[45] T. Ishihara, K. M. Kitani, W. C. Ma, H. Takagi, and C. Asakawa. "Recognizing hand-object interactions in wearable camera videos." In: *2015 IEEE International Conference on Image Processing (ICIP)*. 2015, pp. 1349–1353. DOI: 10.1109/ICIP.2015.7351020.

[46] Jonathan Kelly and Gaurav S Sukhatme. "Visual-inertial sensor fusion: Localization, mapping and sensor-to-sensor self-calibration." In: *The International Journal of Robotics Research* 30.1 (2011), pp. 56–79.

[47] Adil Mehmood Khan, Y-K Lee, SY Lee, and T-S Kim. "Human activity recognition via an accelerometer-enabled-smartphone using kernel discriminant analysis." In: *Future Information Technology (FutureTech), 2010 5th International Conference on*. IEEE. 2010, pp. 1–6.

[48] Michael Kipp. "ANVIL - A generic annotation tool for multimodal dialogue." In: *Seventh European Conference on Speech Communication and Technology*. ISCA, 2001, pp. 1367–1370.

[49] Christian Krupitzer, Timo Sztyler, Janick Edinger, Martin Breitbach, Heiner Stuckenschmidt, and Christian Becker. "Hips do lie! A position-aware mobile fall detection system." In: *2018 IEEE International Conference on Pervasive Computing and Communications (PerCom)*. IEEE. 2018, pp. 1–10.

[50] Amit Kumar, Kristina Yordanova, Thomas Kirste, and Mohit Kumar. "Combining off-the-shelf Image Classifiers with Transfer Learning for Activity Recognition." In: *Proceedings of the 5th international Workshop on Sensor-based Activity Recognition and Interaction*. ACM. 2018, p. 15.

[51] Jennifer R. Kwapisz, Gary M. Weiss, and Samuel A. Moore. "Activity Recognition Using Cell Phone Accelerometers." In: *SIGKDD Explorations Newsletter* 12.2 (2011), pp. 74–82. DOI: 10.1145/1964897.1964918.

[52] O. D. Lara and M. A. Labrador. "A Survey on Human Activity Recognition using Wearable Sensors." In: *IEEE Communications Surveys Tutorials* 15.3 (2013), pp. 1192–1209. DOI: 10.1109/SURV.2012.110112.00192.

[53] M Powell Lawton and Elaine M Brody. "Assessment of older people: Self-maintaining and instrumental activities of daily living." In: *The gerontologist* 9.3_Part_1 (1969), pp. 179–186.

[54] Jinna Lei, Xiaofeng Ren, and Dieter Fox. "Fine-grained kitchen activity recognition using rgb-d." In: *Proceedings of the 2012 ACM Conference on Ubiquitous Computing*. ACM. 2012, pp. 208–211.

[55] Xingyan Li, Ian Yen-Hung Chen, Stephen Thomas, and Bruce A MacDonald. "Using Kinect for monitoring warehouse order picking operations." In: *Proceedings of Australasian Conference on Robotics and Automation*. Vol. 15. 2012, pp. 1–7.

[56] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. "Feature pyramid networks for object detection." In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, pp. 2117–2125.

[57] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. "Microsoft COCO: Common Objects in Context." In: *Computer Vision (ECCV)*. Springer International Publishing, 2014, pp. 740–755.

[58] Ce Liu, William T Freeman, Edward H Adelson, and Yair Weiss. "Human-assisted motion annotation." In: *IEEE Conference on Computer Vision and Pattern Recognition*. IEEE Computer Society, 2008, pp. 1–8.

[59] Leo Louis. "working principle of Arduino and u sing it." In: *International Journal of Control, Automation, Communication and Systems (IJCACS)* 1.2 (2016), pp. 21–29.

[60] Takuya Maekawa, Yutaka Yanagisawa, Yasue Kishino, Katsuhiko Ishiguro, Koji Kamei, Yasushi Sakurai, and Takeshi Okadome. "Object-based activity recognition with heterogeneous sensors on wrist." In: *International Conference on Pervasive Computing*. Springer. 2010, pp. 246–264.

[61] Jenny Margarito, Rim Helaoui, Anna M Bianchi, Francesco Sartor, and Alberto G Bonomi. "User-Independent Recognition of Sports Activities From a Single Wrist-Worn Accelerometer: A Template-Matching-Based Approach." In: *IEEE Transactions on Biomedical Engineering* 63.4 (2016), pp. 788–796.

[62] Christine F Martindale, Florian Hoenig, Christina Strohrmann, and Bjoern M Eskofier. "Smart Annotation of Cyclic Data Using Hierarchical Hidden Markov Models." In: *Sensors* 17.10 (2017), p. 2328.

[63] Aaron Miller. "Order picking for the 21st Century – Voice vs. Scanning Technology." white paper. 2004.

[64] Mladen Milošević, Michael T Shrove, and Emil Jovanov. "Applications of smartphones for ubiquitous health monitoring and wellbeing management." In: *JITA-Journal Of Information Technology and Aplications* 1.1 (2011).

[65] Elisa Morganti, Leonardo Angelini, Andrea Adami, Denis Lalanne, Leandro Lorenzelli, and Elena Mugellini. "A Smart Watch with Embedded Sensors to Recognize Objects, Grasps and Forearm Gestures." In: *Procedia Engineering* 41 (2012), pp. 1169–1175. DOI: http://dx.doi.org/10.1016/j.proeng.2012.07.297.

[66] Lindasalwa Muda, Mumtaj Begam, and Irraivan Elamvazuthi. "Voice recognition algorithms using mel frequency cepstral coefficient (MFCC) and dynamic time warping (DTW) techniques." In: *arXiv preprint arXiv:1003.4083* (2010).

[67] Meinard Müller. *Fundamentals of Music Processing: Audio, Analysis, Algorithms, Applications*. 978-3-319-21945-5. Springer, 2015.

[68] Naoko Muramatsu and Hiroko Akiyama. "Japan: super-aging society preparing for the future." In: *The Gerontologist* 51.4 (2011), pp. 425–432.

[69] Yunyoung Nam, Seungmin Rho, and Chulung Lee. "Physical Activity Recognition Using Multiple Sensors Embedded in a Wearable Device." In: *ACM Transactions on Embedded Computing Systems* 12.2 (2013), 26:1–26:14. DOI: 10.1145/2423636.2423644.

[70] Thi-Hoa-Cuc Nguyen, Jean-Christophe Nebel, and Francisco Florez-Revuelta. "Recognition of activities of daily living with egocentric vision: A review." In: *Sensors* 16.1 (2016), p. 72. DOI: 10.3390/s16010072.

[71] Francisco Javier Ordóñez and Daniel Roggen. "Deep convolutional and LSTM recurrent neural networks for multimodal wearable activity recognition." In: *Sensors* 16.1 (2016), p. 115.

[72] Z. Palotai, M. Láng, A. Sárkány, Z. Tősér, D. Sonntag, T. Toyama, and A. Lőrincz. "LabelMovie: Semi-supervised machine annotation tool with quality assurance and crowd-sourcing options for videos." In: *12th International Workshop on Content-Based Multimedia Indexing*. IEEE Computer Society, 2014, pp. 1–4. DOI: 10.1109/CBMI.2014.6849850.

[73] H. Pirsiavash and D. Ramanan. "Detecting activities of daily living in first-person camera views." In: *2012 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE Computer Society, 2012, pp. 2847–2854. DOI: 10.1109/CVPR.2012.6248010.

[74] S. J. Preece, J. Y. Goulermas, L. P. J. Kenney, and D. Howard. "A Comparison of Feature Extraction Methods for the Classification of Dynamic Activities From Accelerometer Data." In: *IEEE Transactions on Biomedical Engineering* 56.3 (2009), pp. 871–879. DOI: 10.1109/TBME.2008.2006190.

[75] Valentin Radu, Nicholas D Lane, Sourav Bhattacharya, Cecilia Mascolo, Mahesh K Marina, and Fahim Kawsar. "Towards multimodal deep learning for activity recognition on mobile devices." In: *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct*. 2016, pp. 185–188.

[76] Valentin Radu, Catherine Tong, Sourav Bhattacharya, Nicholas D Lane, Cecilia Mascolo, Mahesh K Marina, and Fahim Kawsar. "Multimodal deep learning for activity and context recognition." In: *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 1.4 (2018), pp. 1–27.

[77] Reza Rawassizadeh, Blaine A. Price, and Marian Petre. "Wearables: Has the Age of Smartwatches Finally Arrived?" In: *Communications of the ACM* 58.1 (2014), pp. 45–47. DOI: 10.1145/2629633.

[78] Daniele Riboni, Timo Sztyler, Gabriele Civitarese, and Heiner Stuckenschmidt. "Unsupervised Recognition of Interleaved Activities of Daily Living through Ontological and Probabilistic Reasoning." In: *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing*. ACM, 2016, pp. 1–12. DOI: 10.1145/2971648.2971691.

[79] Alina Roitberg, Nikhil Somani, Alexander Perzylo, Markus Rickert, and Alois Knoll. "Multimodal human activity recognition for industrial manufacturing processes in robotic workcells." In: *Proceedings of the 2015 ACM on International Conference on Multimodal Interaction*. 2015, pp. 259–266.

[80] Nirmalya Roy, Archan Misra, and Diane Cook. "Ambient and smartphone sensor assisted ADL recognition in multi-inhabitant smart environments." In: *Journal of ambient intelligence and humanized computing* 7.1 (2016), pp. 1–19.

[81] Rubén San-Segundo, Juan Manuel Montero, Roberto Barra-Chicote, Fernando Fernández, and José Manuel Pardo. "Feature extraction from smartphone inertial signals for human activity segmentation." In: *Signal Processing* 120 (2016), pp. 359–372.

[82] S. Song, V. Chandrasekhar, B. Mandal, L. Li, J.-H. Lim, G. Sateesh Babu, P. San, and N.-M. Cheung. "Multimodal Multi-Stream Deep Learning for Egocentric Activity Recognition." In: *2016 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. IEEE Computer Society, 2016, pp. 378–385. DOI: 10.1109/CVPRW.2016.54.

[83] S. Song, N. M. Cheung, V. Chandrasekhar, B. Mandal, and J. Liri. "Egocentric activity recognition with multimodal fisher vector." In: *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE Computer Society, 2016, pp. 2717–2721. DOI: 10.1109/ICASSP.2016.7472171.

[84] Ekaterina H Spriggs, Fernando De La Torre, and Martial Hebert. "Temporal segmentation and activity classification from first-person sensing." In: *Computer Vision and Pattern Recognition Workshops, 2009. CVPR Workshops 2009. IEEE Computer Society Conference On*. IEEE. 2009, pp. 17–24. DOI: 10.1109/CVPRW.2009.5204354.

[85] Allan Stisen, Henrik Blunck, Sourav Bhattacharya, Thor Siiger Prentow, Mikkel Baun Kjærgaard, Anind Dey, Tobias Sonne, and Mads Møller Jensen. "Smart devices are different: Assessing and mitigatingmobile sensing heterogeneities for activity recognition." In: *Proceedings of the 13th ACM conference on embedded networked sensor systems*. 2015, pp. 127–140.

[86] S Szewcyzk, K Dwan, B Minor, B Swedlove, and D Cook. "Annotating smart environment sensor data for activity learning." In: *Technology and Health Care* 17.3 (2009), pp. 161–169.

[87] Timo Sztyler and Heiner Stuckenschmidt. "On-body Localization of Wearable Devices: An Investigation of Position-Aware Activity Recognition." In: *2016 IEEE International Conference on Pervasive Computing and Communications (PerCom)*. IEEE Computer Society, 2016, pp. 1–9. DOI: 10.1109/PERCOM.2016.7456521.

[88] Fernando De la Torre, Jessica Hodgins, Adam Bargteil, Xavier Martin, Justin Macey, Alex Collado, and Pep Beltran. "Guide to the carnegie mellon university multimodal activity (cmu-mmac) database." In: *Robotics Institute* (2008), p. 135.

[89] TS Vaughan. "The effect of warehouse cross aisles on order picking efficiency." In: *International Journal of Production Research* 37.4 (1999), pp. 881–897.

[90] Carl Vondrick, Donald Patterson, and Deva Ramanan. "Efficiently scaling up crowdsourced video annotation." In: *International Journal of Computer Vision* 101.1 (2013), pp. 184–204.

[91] Hao Wang, Daqing Zhang, Yasha Wang, Junyi Ma, Yuxiang Wang, and Shengjie Li. "RT-Fall: A Real-Time and Contactless Fall Detection System with Commodity WiFi Devices." In: *IEEE Trans. Mob. Comput.* 16.2 (2017), pp. 511–526.

[92] G. M. Weiss, J. L. Timko, C. M. Gallagher, K. Yoneda, and A. J. Schreiber. "Smartwatch-based activity recognition: A machine learning approach." In: *2016 IEEE-EMBS International Conference on Biomedical and Health Informatics (BHI)*. IEEE Computer Society, 2016, pp. 426–429. DOI: 10.1109/BHI.2016.7455925.

[93] Gary M Weiss, Jeffrey W Lockhart, Tony T Pulickal, Paul T McHugh, Isaac H Ronan, and Jessica L Timko. "Actitracker: a smartphone-based activity recognition system for improving health and well-being." In: *2016 IEEE International Conference on Data Science and Advanced Analytics (DSAA)*. IEEE. 2016, pp. 682–688.

[94] Daniel H Wilson. *Assistive intelligent environments for automatic health monitoring*. Carnegie Mellon University, 2005.

[95] J. Windau and L. Itti. "Situation awareness via sensor-equipped eyeglasses." In: *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE Computer Society, 2013, pp. 5674–5679. DOI: 10.1109/IROS.2013.6697178.

[96] Michael Wölfle and Willibald A Günthner. "Wearable RFID in order picking systems." In: *Smart Objects: Systems, Technologies and Applications, Proceedings of RFID SysTech 2011 7th European Workshop on*. VDE. 2011, pp. 1–6.

[97] Oliver J Woodman. *An introduction to inertial navigation*. Tech. rep. University of Cambridge, Computer Laboratory, 2007.

[98] Di Wu, Lionel Pigou, Pieter-Jan Kindermans, Nam Do-Hoang Le, Ling Shao, Joni Dambre, and Jean-Marc Odobez. "Deep dynamic neural networks for multimodal gesture segmentation and recognition." In: *IEEE transactions on pattern analysis and machine intelligence* 38.8 (2016), pp. 1583–1597.

[99] J. Wu, A. Osuntogun, T. Choudhury, M. Philipose, and J. M. Rehg. "A Scalable Approach to Activity Recognition based on Object Use." In: *2007 IEEE 11th International Conference on Computer Vision*. IEEE Computer Society, 2007, pp. 1–8. DOI: 10.1109/ICCV.2007.4408865.

[100] Jun Yang. "Toward physical activity diary: motion recognition using simple acceleration features with mobile phones." In: *Proceedings of the 1st international workshop on Interactive multimedia for consumer electronics*. ACM. 2009, pp. 1–10.

[101] Kristina Yordanova, Frank Krüger, and Thomas Kirste. "Providing semantic annotation for the cmu grand challenge dataset." In: *2018 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops)*. IEEE. 2018, pp. 579–584.

[102] Guopeng Zhang and Massimo Piccardi. "Structural SVM with partial ranking for activity segmentation and classification." In: *IEEE Signal Processing Letters* 22.12 (2015), pp. 2344–2348.

[103] Barret Zoph, Vijay Vasudevan, Jonathon Shlens, and Quoc V Le. "Learning transferable architectures for scalable image recognition." In: *arXiv preprint arXiv:1707.07012* 2.6 (2017).

[104] Han Zou, Jianfei Yang, Hari Prasanna Das, Huihan Liu, Yuxun Zhou, and Costas J Spanos. "Wifi and vision multimodal learning for accurate and robust device-free human activity recognition." In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*. 2019, pp. 0–0.