(health & activity) sensors

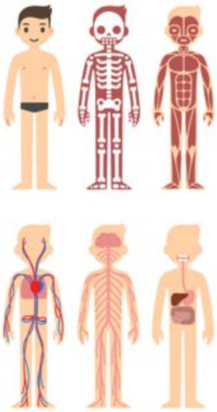
mobile health & activity monitoring
spring term 2023 ETH Zürich
prof. christian holz siplab

“health” :=

the state of wellness of a human

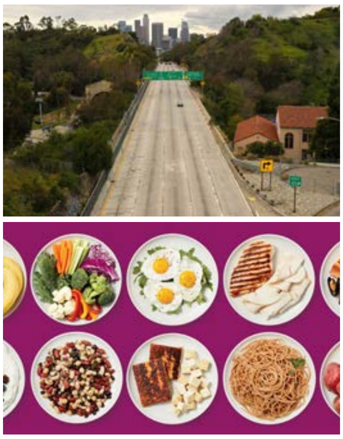
state of wellness

physiological



mental

environmental



genetical

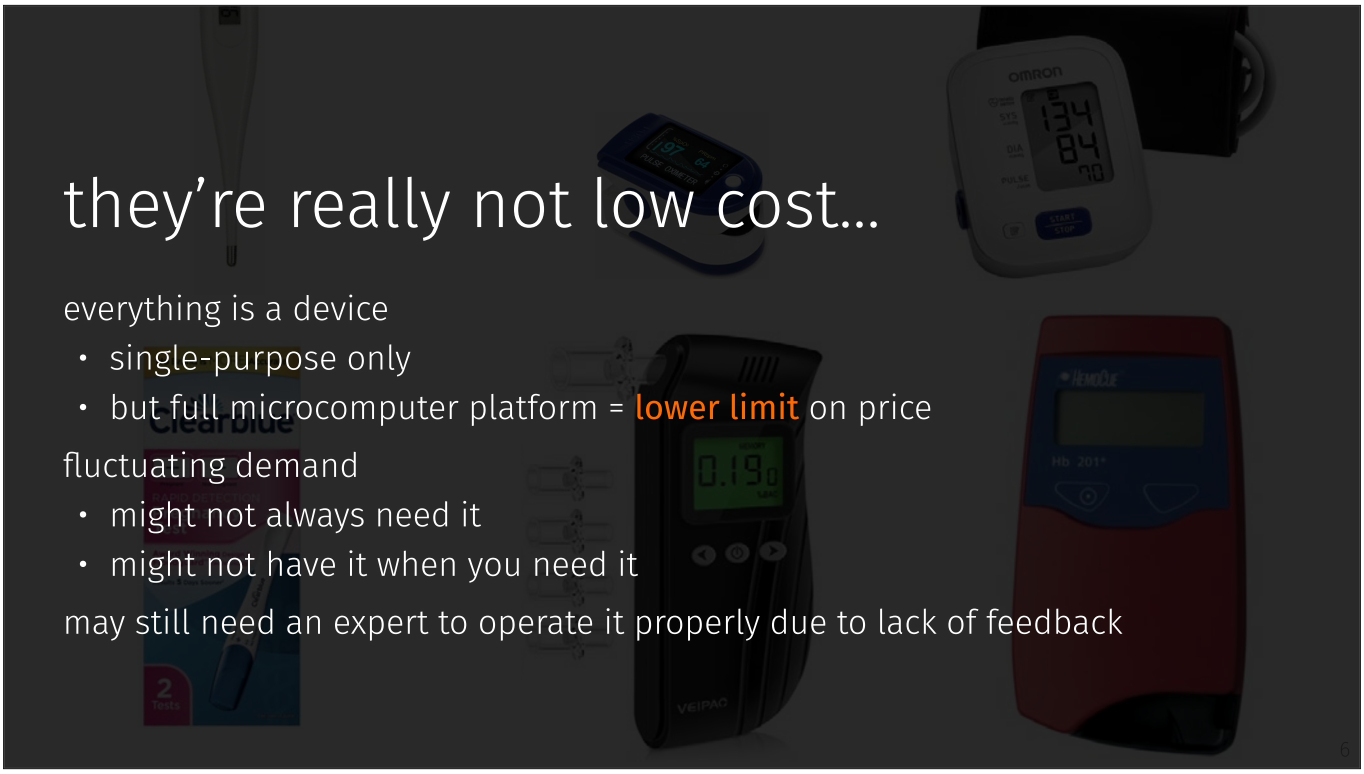
today’s low-cost health monitors

thermometer – $7

pulse oximeter – $15

BP monitor – $25

today’s low-cost health monitors



they’re really not low cost…

everything is a device

- single-purpose only
- but full microcomputer platform = lower limit on price

fluctuating demand

- might not always need it
- might not have it when you need it

may still need an expert to operate it properly due to lack of feedback

but components are a commodity

display

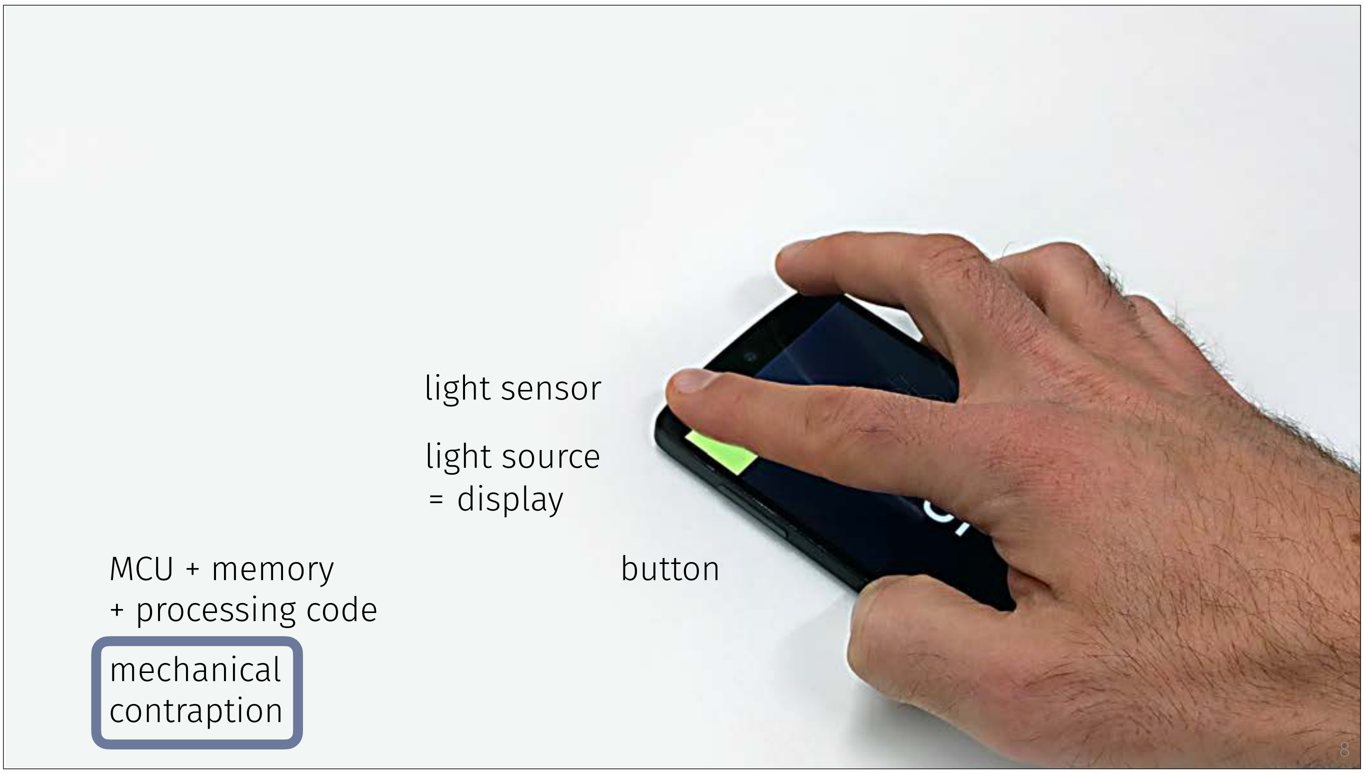
mechanical
contraption

button

MCU + memory
+ processing code

light source

light sensor



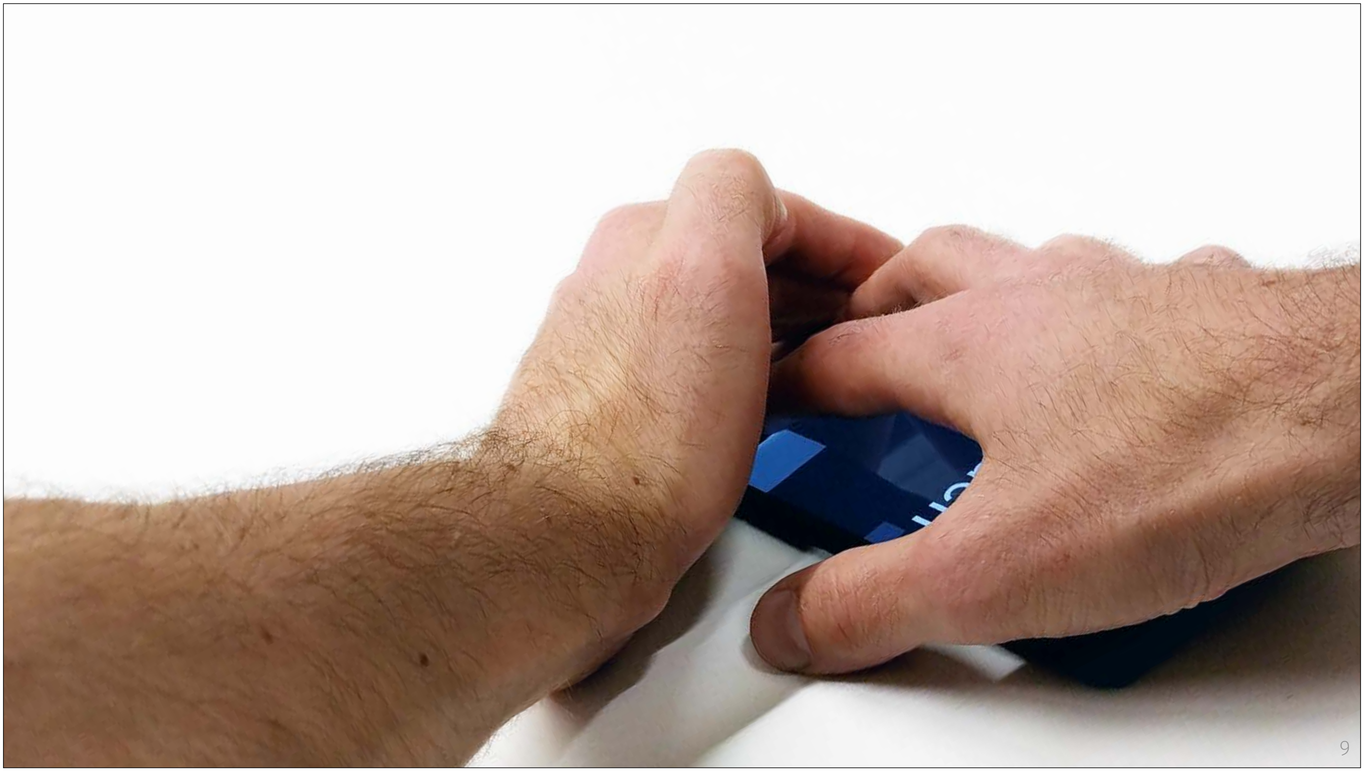
light sensor

light source
= display

MCU + memory
+ processing code

button

mechanical
contraption



smartphone:
low-cost vehicle for medical devices

connectivity + interactivity
➪ telemedicine

sensors + on-board processing
➪ digitization

interactivity
➪ assistance in self-care

checking email + playing games + photography
➪ more customers beyond medicine
➪ economy of scale

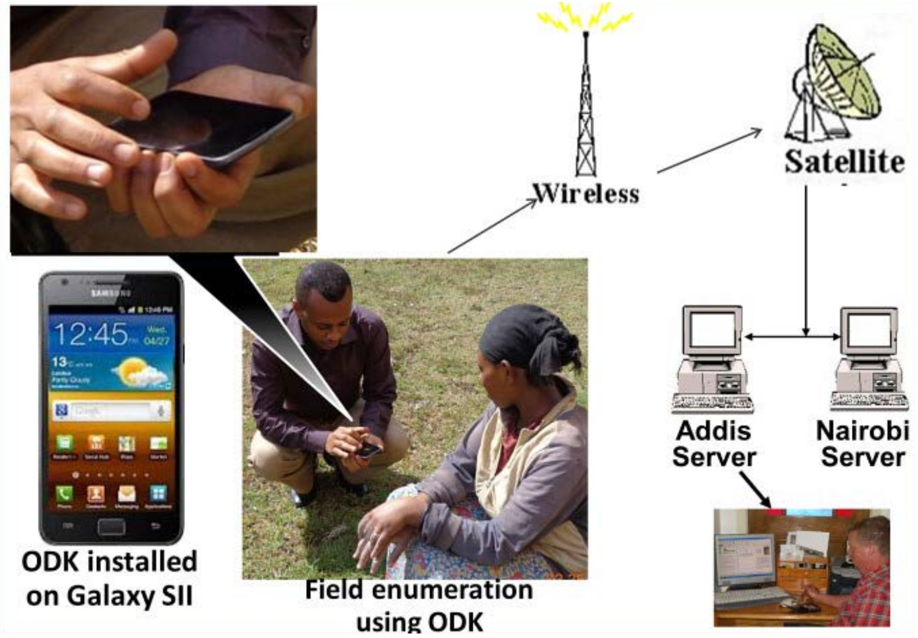
➪ all using the same approach

telemedicine

user entering
  detailed information

connectivity

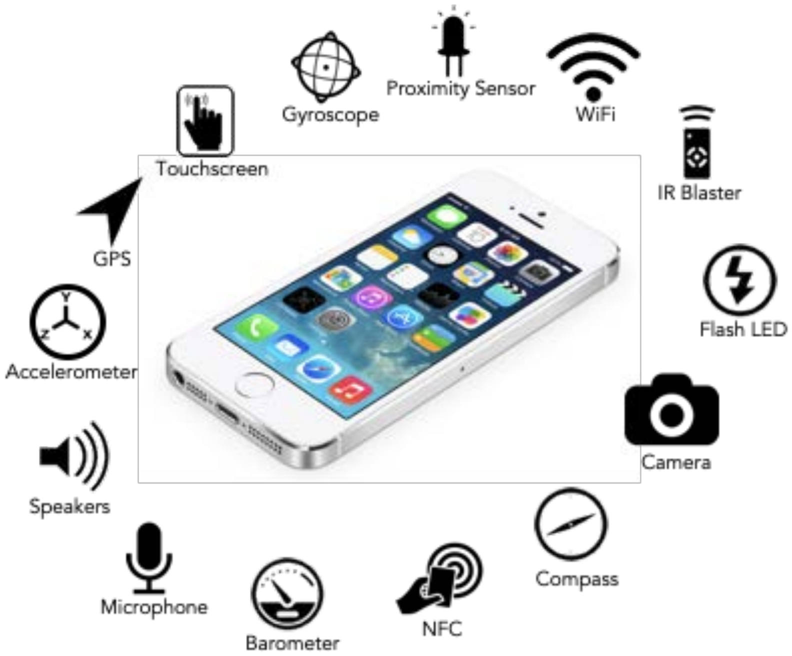
backend + expert



digitization & interactive assistance

using the phone's processing power + on-phone processing to
digitize, store, transmit the data from medical sensors

phone itself as a medical actuator+sensor

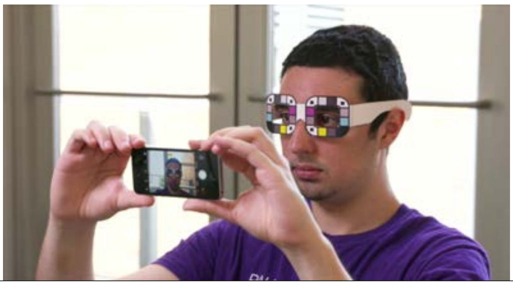


passive augmentation

to compensate for shortcomings

aids for color correction

(assistance to) direct/excite the signal



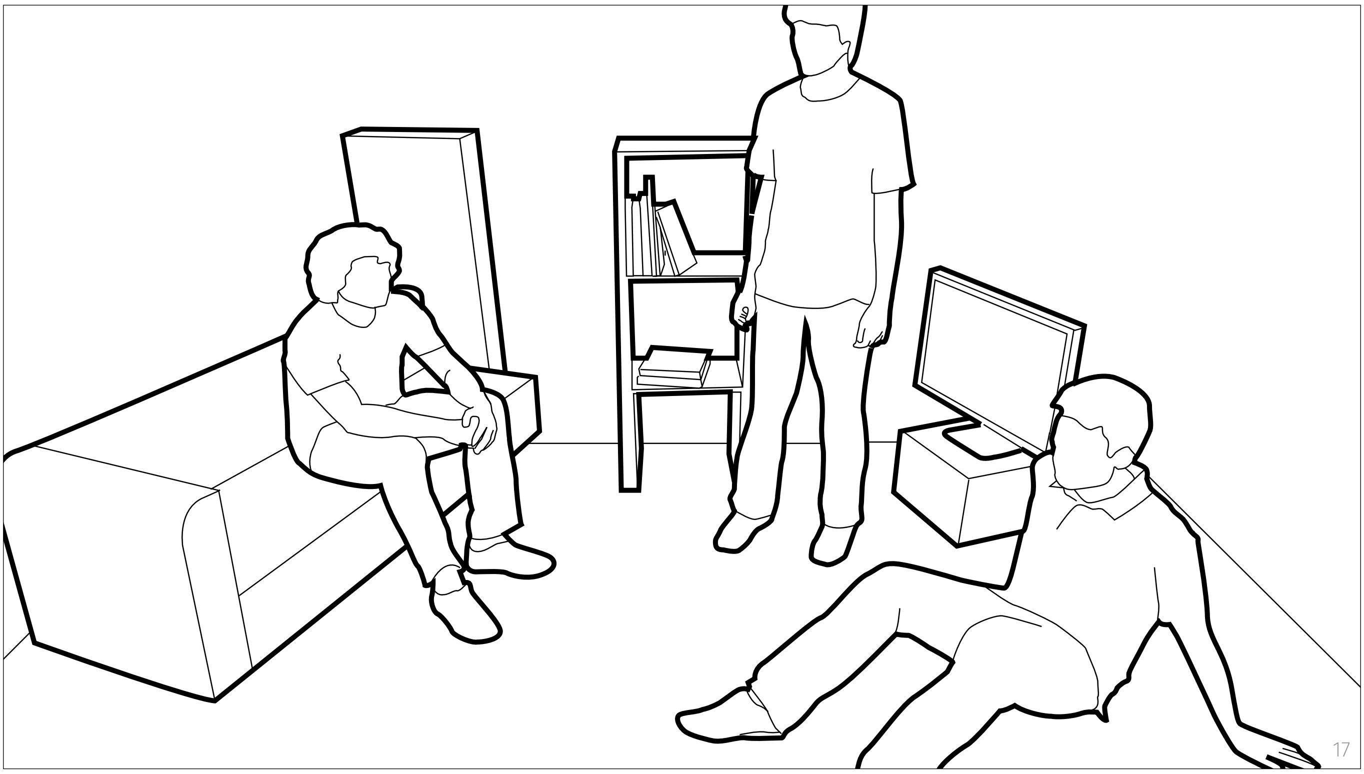
multimodal sensing

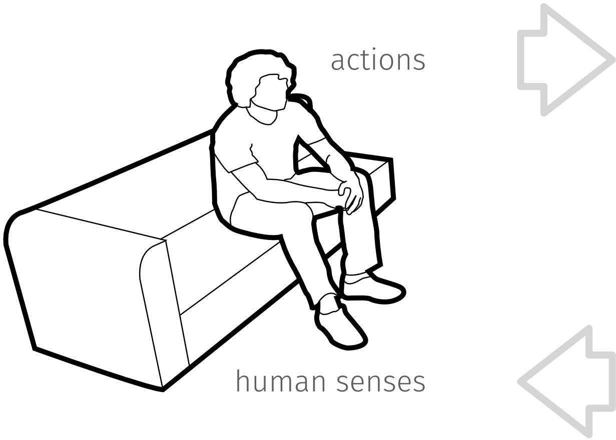
accelerometer:
detects heart beat

camera:
detects pulse wave in fingertip

temporal offset yields pulse-transit time

sensors as
the human-computer **interface**





actions

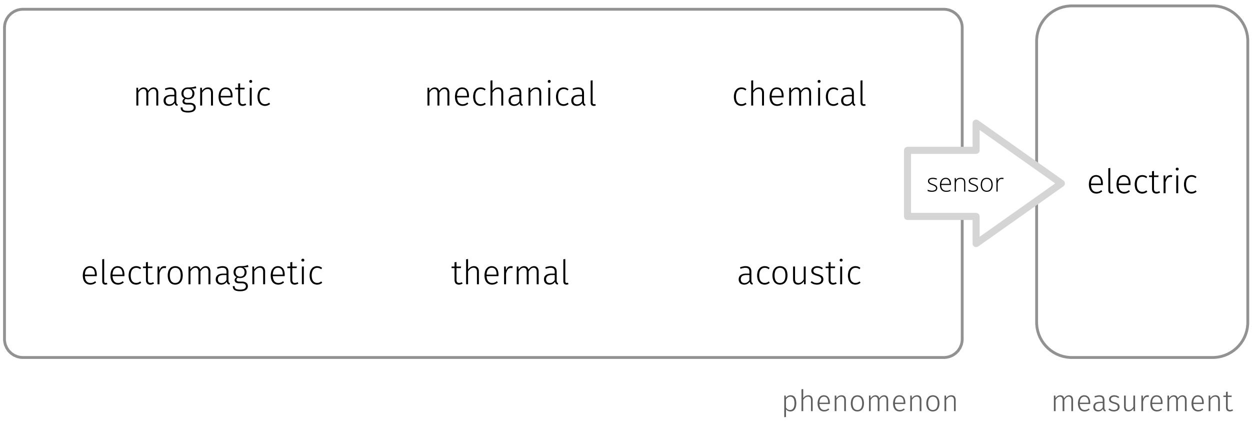
human senses

sensors

device that detects or measures a physical property and records, indicates, or otherwise responds to it electrically

displays

what is a sensor?



magnetic mechanical chemical

electromagnetic thermal acoustic

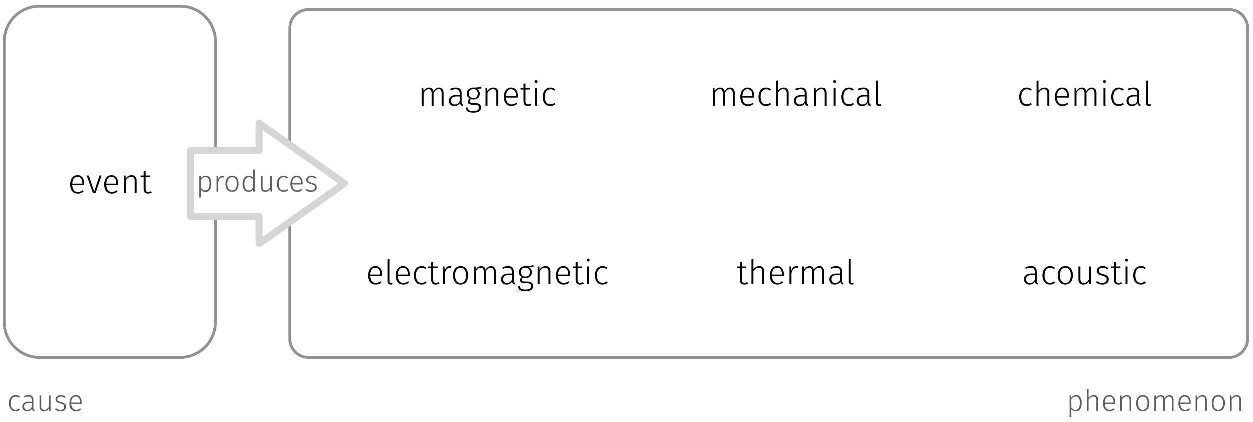
sensor

electric

phenomenon

measurement

what is a sensor?



event

produces

magnetic mechanical chemical

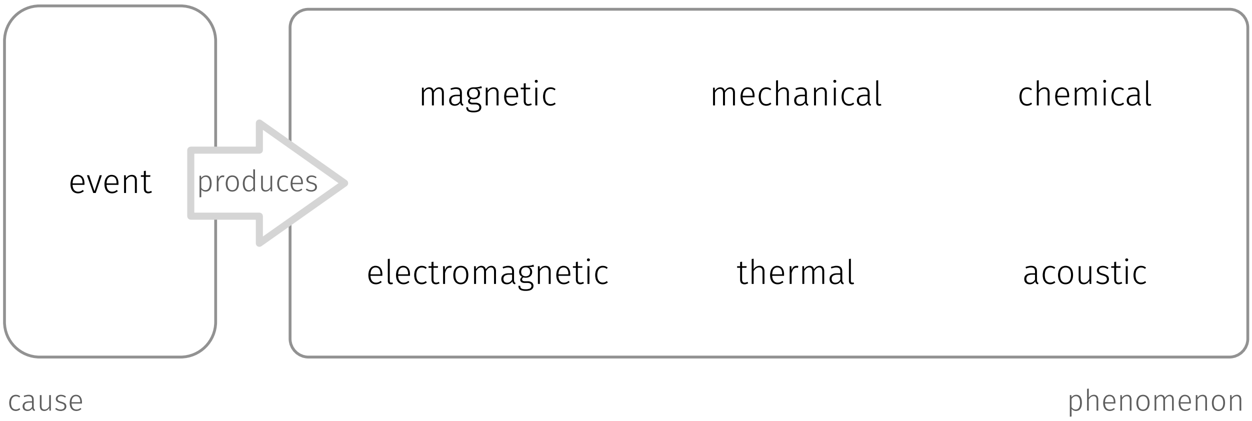
electromagnetic thermal acoustic

cause

phenomenon

what could they be in a smart home scenario?
30-second brainstorming—talk to your neighbor!

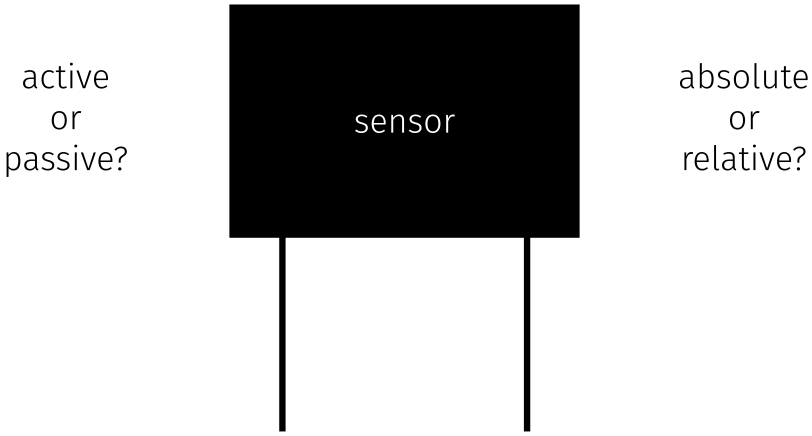
what is a sensor?



what could they be in a **wearable scenario?**

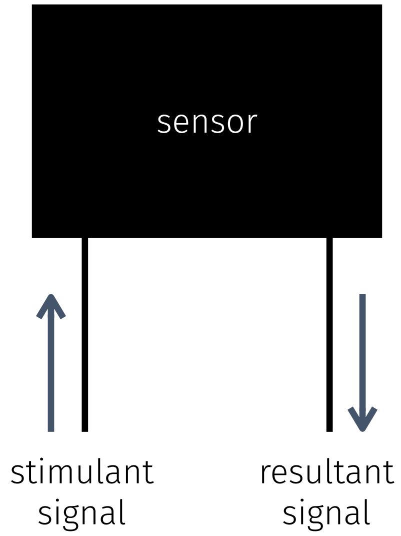
30-second brainstorming—talk to your neighbor!

sensor characteristics

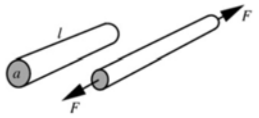


sensor types: active

- resistance
- capacitance
- inductance
- hall effect



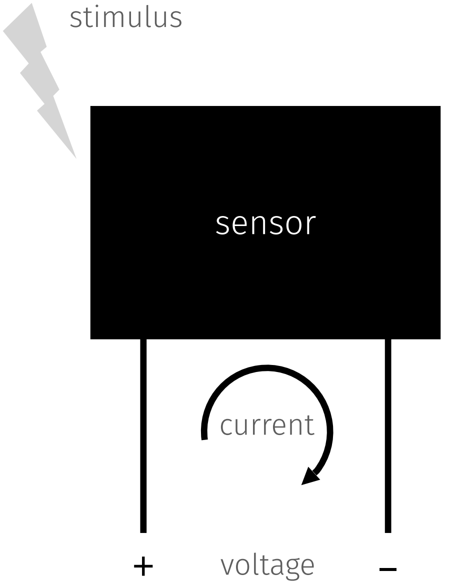
stretch sensor



require external power for operation

sensor types: passive

- piezoelectric
- pyroelectric
- thermoelectric
- photoelectric
- faraday’s law

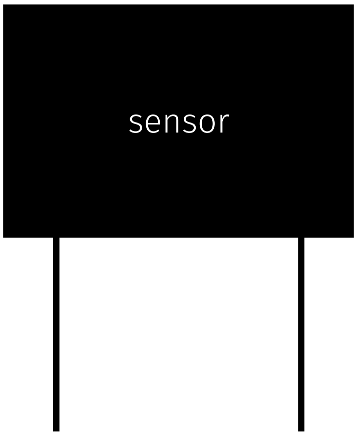


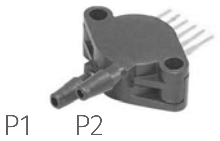
does not require external power

sensor types: references

absolute sensors

relative sensors

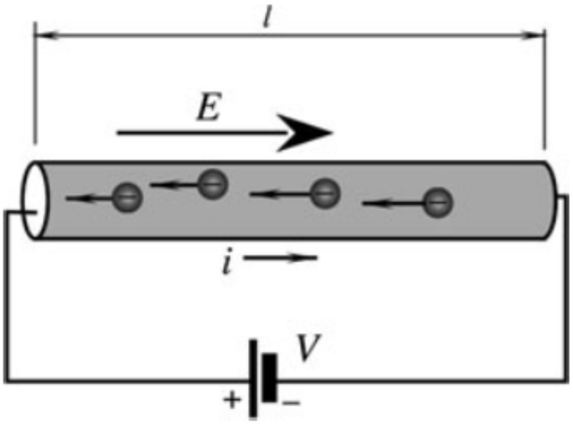




atmospheric pressure sensor

differential pressure sensor

resistance (R)



length of conductor

$$R = \frac{V}{i} = \rho\frac{l}{a}$$

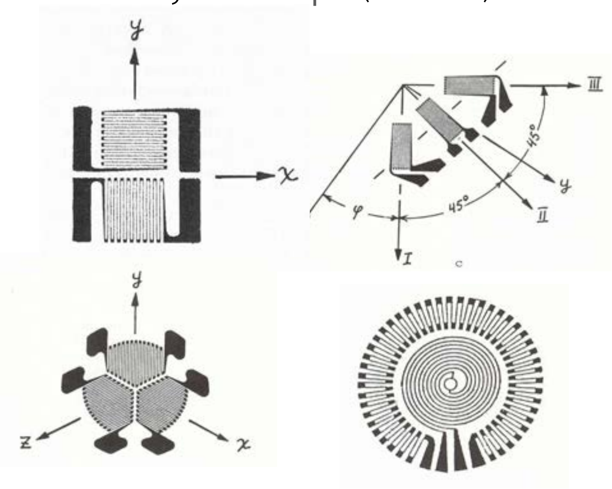
cross sectional area

specific resistivity

“bad” resistor ➪ good resistive sensor

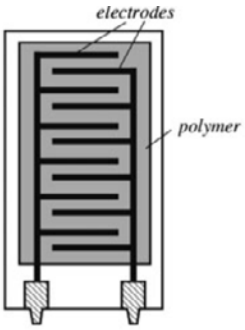
resistance: applications

vary the shape (l and a)



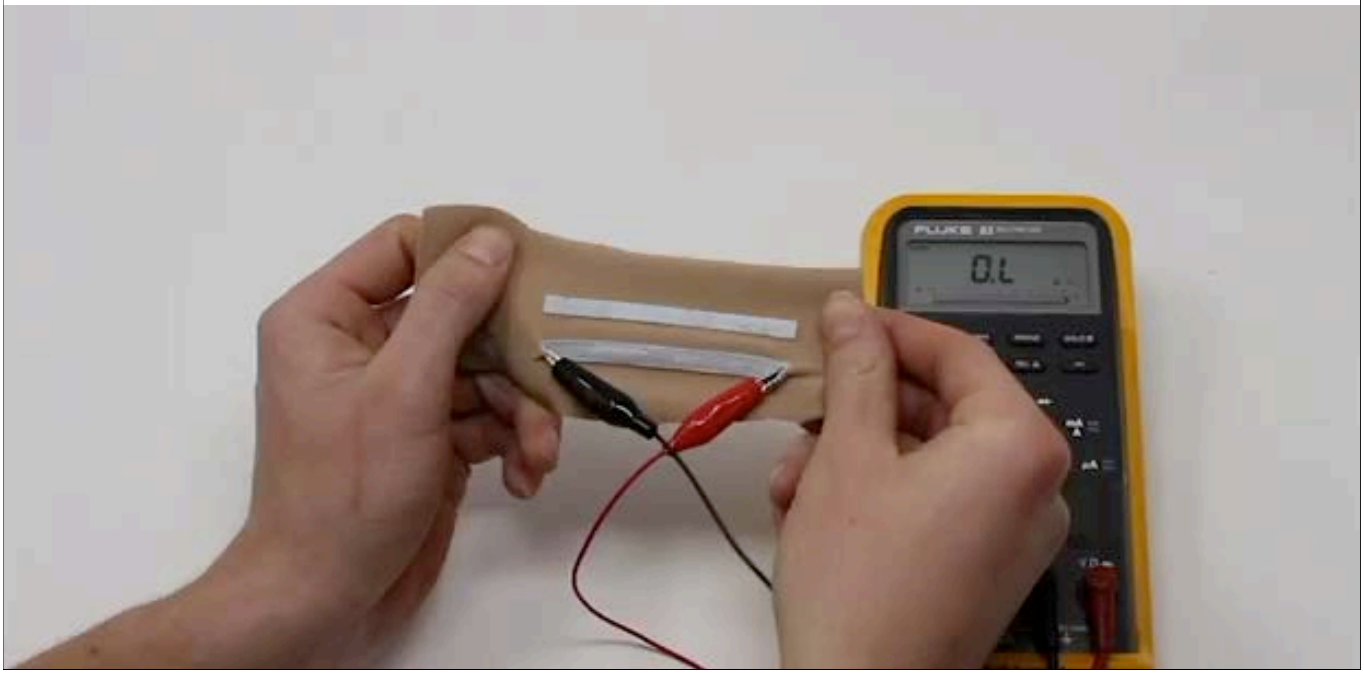
strain gauges

vary the resistivity (ρ)

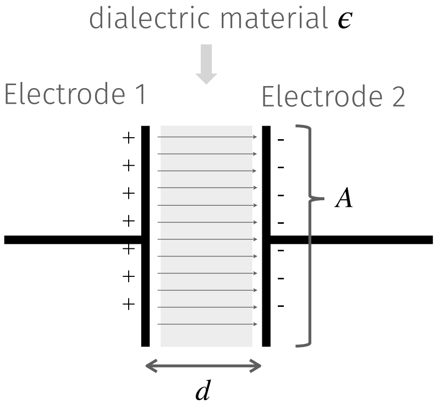


thermo-resistive elements
moisture sensors

resistance: wire inside silicone ➪ stretch



capacitance (C)


dialectric material ϵ

Electrode 1

Electrode 2

area of electrode

$$C = \epsilon \frac{A}{d}$$

electrode spacing

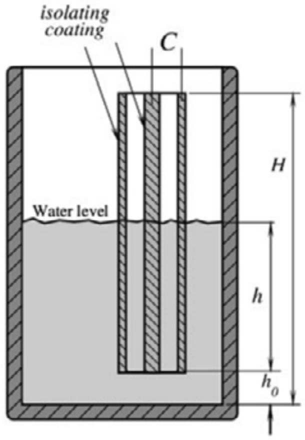
dielectric constant

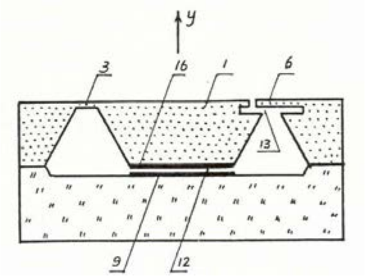
capacitance: applications

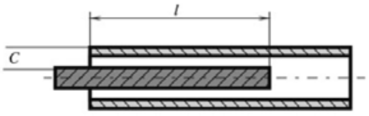
vary the dielectric (ε)

vary the spacing (d)

vary the area (A)



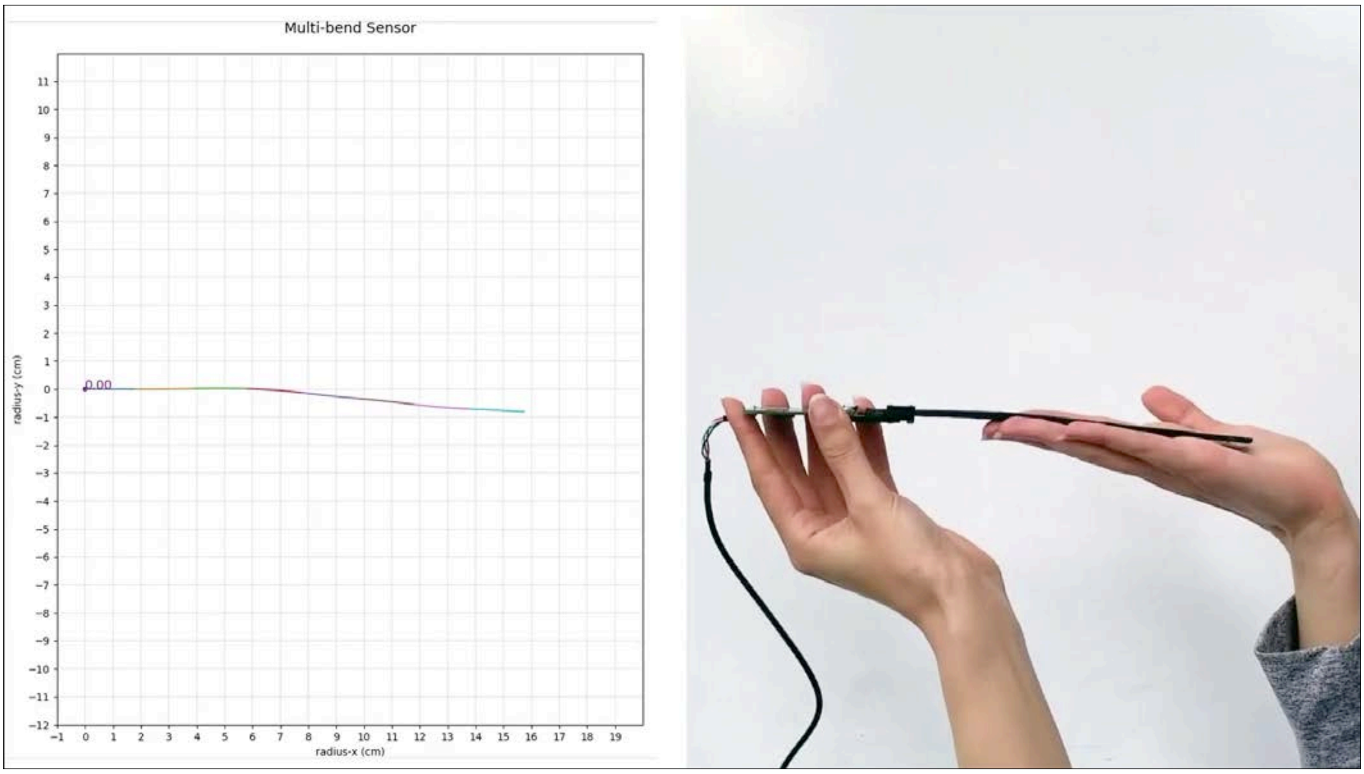




water level sensor

MEMS (microelectromechanical systems) accelerometer

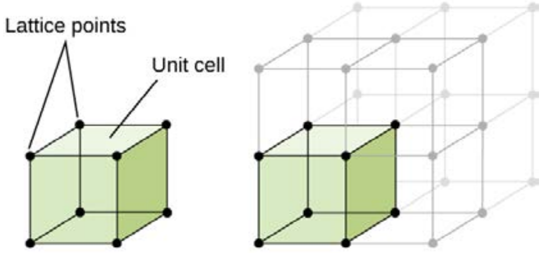
slider



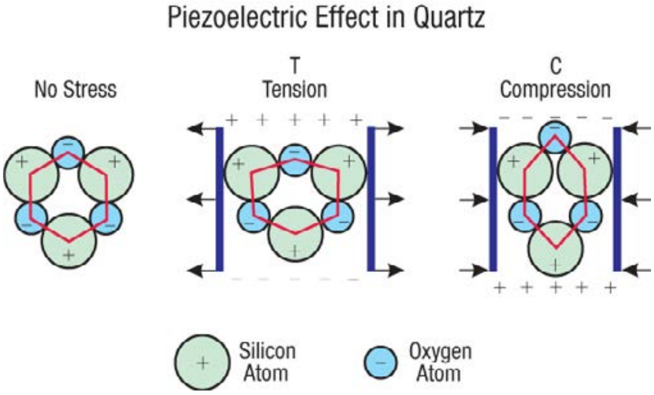
ShArc Prototype

[Shark, CHI '20]

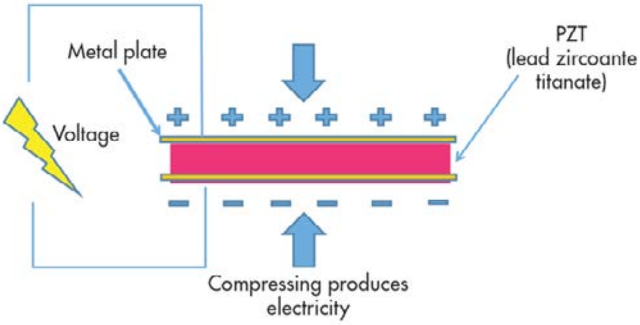
piezoelectric: mechanical to electrical



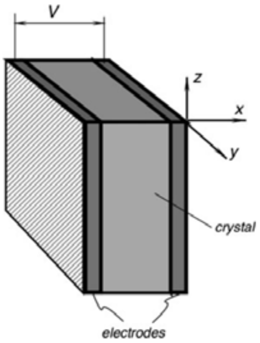
Quartz crystal



piezoelectric: mechanical to electrical

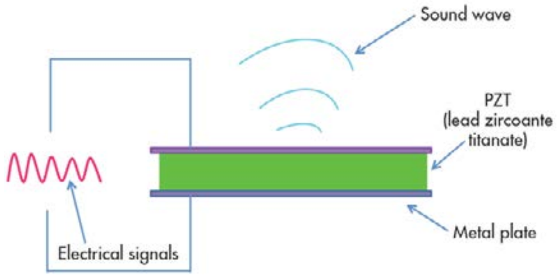


direct piezoelectric effect



sandwich crystal in electrodes
to conduct the electricity

piezoelectric: electrical to mechanical

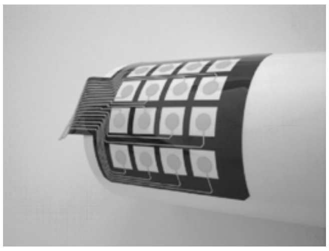


inverse piezoelectric effect

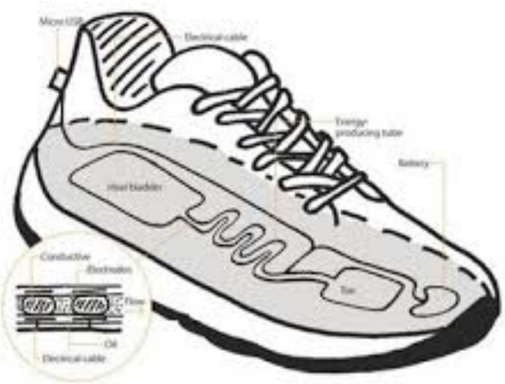
piezoelectric: various forms

crystal

semi-crystalline polymer (PVDF)

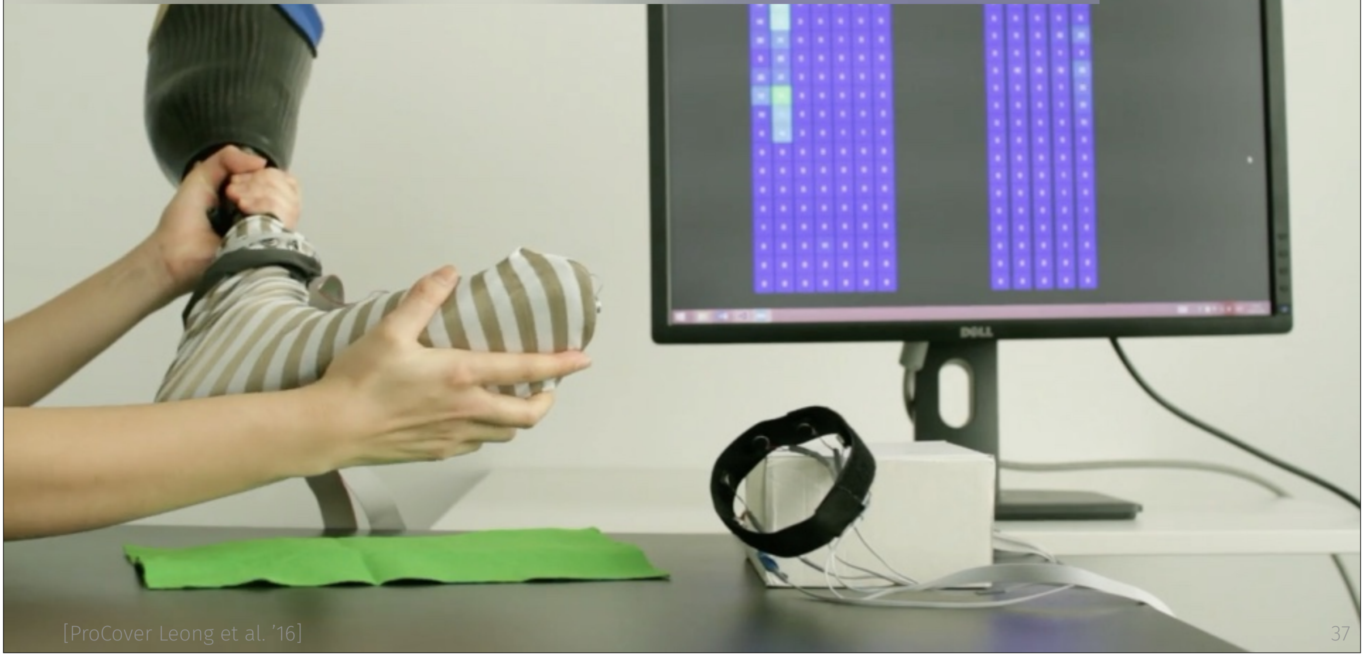


surface acoustics sonar



flexible force sensing

piezoresistive: mechanical to ΔR



[ProCover Leong et al. '16]

proCover
Sensory Augmentation of Prosthetic Limbs Using Smart Textile Covers
DELL

exercise

LilyGo smartwatch

each group gets one

battery-operated wearable band

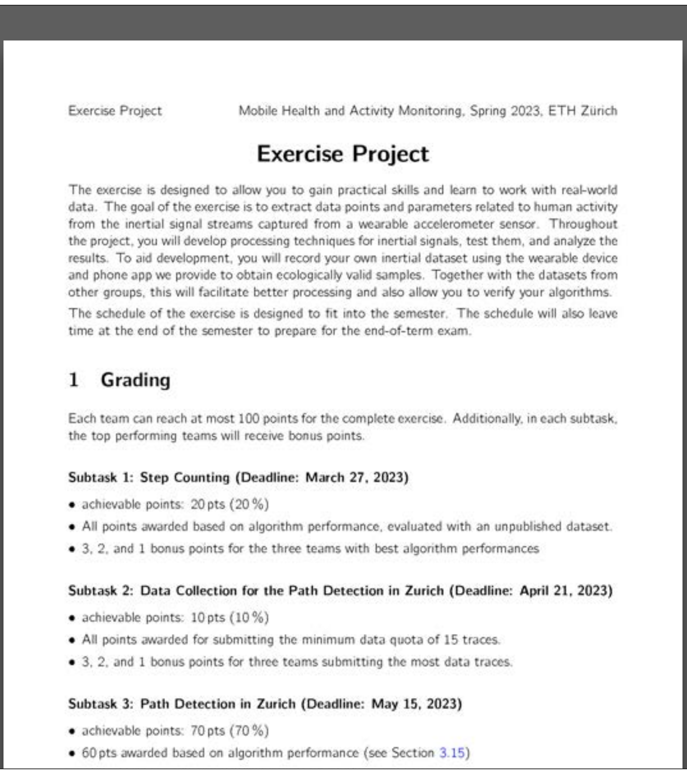
includes sensors

- 3-axis accelerometer
- 3-axis gyroscope
- 3-axis magnetometer
- (sensor) temperature

pairs with your phone

additional sensors

- GPS
- barometer
- …



Exercise Project Mobile Health and Activity Monitoring, Spring 2023, ETH Zürich

Exercise Project

The exercise is designed to allow you to gain practical skills and learn to work with real-world data. The goal of the exercise is to extract data points and parameters related to human activity from the inertial signal streams captured from a wearable accelerometer sensor. Throughout the project, you will develop processing techniques for inertial signals, test them, and analyze the results. To aid development, you will record your own inertial dataset using the wearable device and phone app we provide to obtain ecologically valid samples. Together with the datasets from other groups, this will facilitate better processing and also allow you to verify your algorithms.

The schedule of the exercise is designed to fit into the semester. The schedule will also leave time at the end of the semester to prepare for the end-of-term exam.

1 Grading

Each team can reach at most 100 points for the complete exercise. Additionally, in each subtask, the top performing teams will receive bonus points.

Subtask 1: Step Counting (Deadline: March 27, 2023)

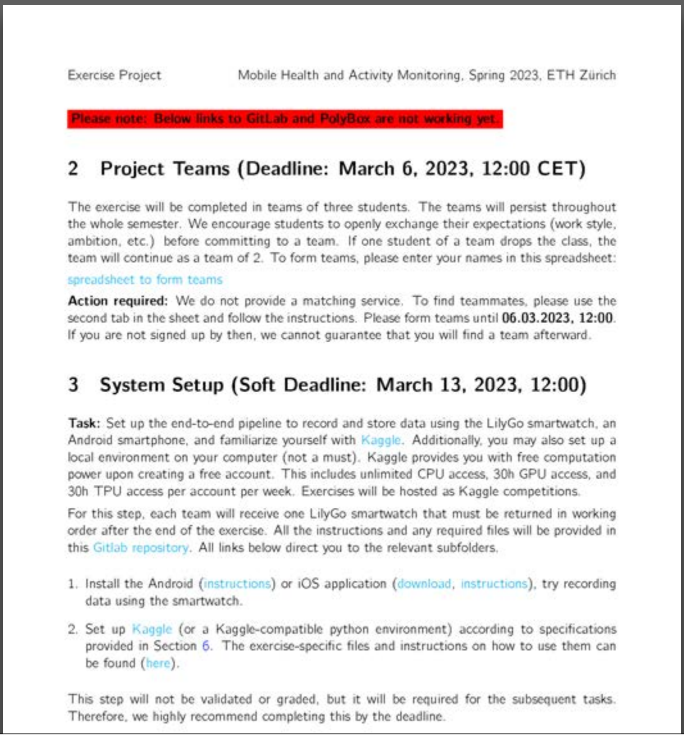
- achievable points: 20 pts (20 %)
- All points awarded based on algorithm performance, evaluated with an unpublished dataset.
- 3, 2, and 1 bonus points for the three teams with best algorithm performances

Subtask 2: Data Collection for the Path Detection in Zurich (Deadline: April 21, 2023)

- achievable points: 10 pts (10 %)
- All points awarded for submitting the minimum data quota of 15 traces.
- 3, 2, and 1 bonus points for three teams submitting the most data traces.

Subtask 3: Path Detection in Zurich (Deadline: May 15, 2023)

- achievable points: 70 pts (70 %)
- 60 pts awarded based on algorithm performance (see Section 3.15)



Exercise Project Mobile Health and Activity Monitoring, Spring 2023, ETH Zürich

Please note: Below links to GitLab and PolyBox are not working yet.

2 Project Teams (Deadline: March 6, 2023, 12:00 CET)

The exercise will be completed in teams of three students. The teams will persist throughout the whole semester. We encourage students to openly exchange their expectations (work style, ambition, etc.) before committing to a team. If one student of a team drops the class, the team will continue as a team of 2. To form teams, please enter your names in this spreadsheet: spreadsheet to form teams

Action required: We do not provide a matching service. To find teammates, please use the second tab in the sheet and follow the instructions. Please form teams until **06.03.2023, 12:00**. If you are not signed up by then, we cannot guarantee that you will find a team afterward.

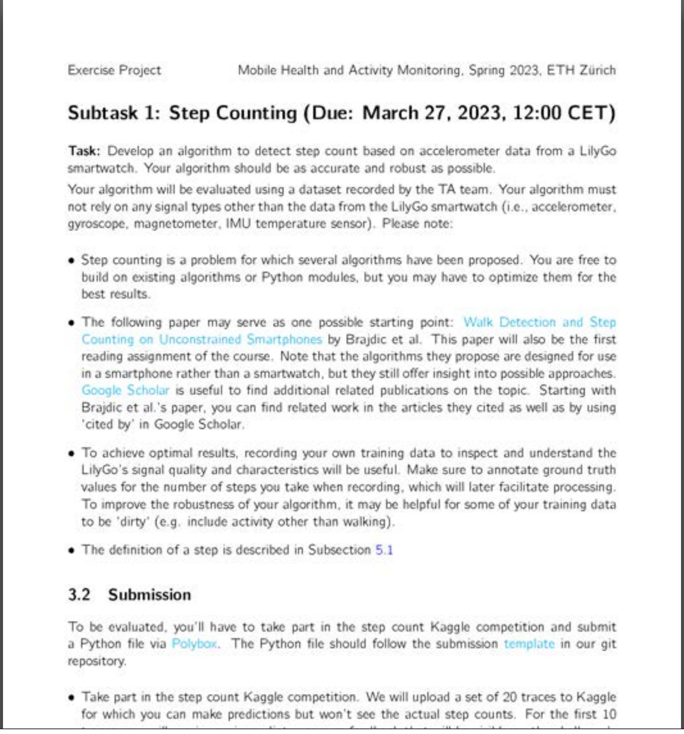
3 System Setup (Soft Deadline: March 13, 2023, 12:00)

Task: Set up the end-to-end pipeline to record and store data using the LilyGo smartwatch, an Android smartphone, and familiarize yourself with Kaggle. Additionally, you may also set up a local environment on your computer (not a must). Kaggle provides you with free computation power upon creating a free account. This includes unlimited CPU access, 30h GPU access, and 30h TPU access per account per week. Exercises will be hosted as Kaggle competitions.

For this step, each team will receive one LilyGo smartwatch that must be returned in working order after the end of the exercise. All the instructions and any required files will be provided in this Gitlab repository. All links below direct you to the relevant subfolders.

1. Install the Android (instructions) or iOS application (download, instructions), try recording data using the smartwatch.
2. Set up Kaggle (or a Kaggle-compatible python environment) according to specifications provided in Section 6. The exercise-specific files and instructions on how to use them can be found (here).

This step will not be validated or graded, but it will be required for the subsequent tasks. Therefore, we highly recommend completing this by the deadline.



Exercise Project Mobile Health and Activity Monitoring, Spring 2023, ETH Zürich

Subtask 1: Step Counting (Due: March 27, 2023, 12:00 CET)

Task: Develop an algorithm to detect step count based on accelerometer data from a LilyGo smartwatch. Your algorithm should be as accurate and robust as possible.

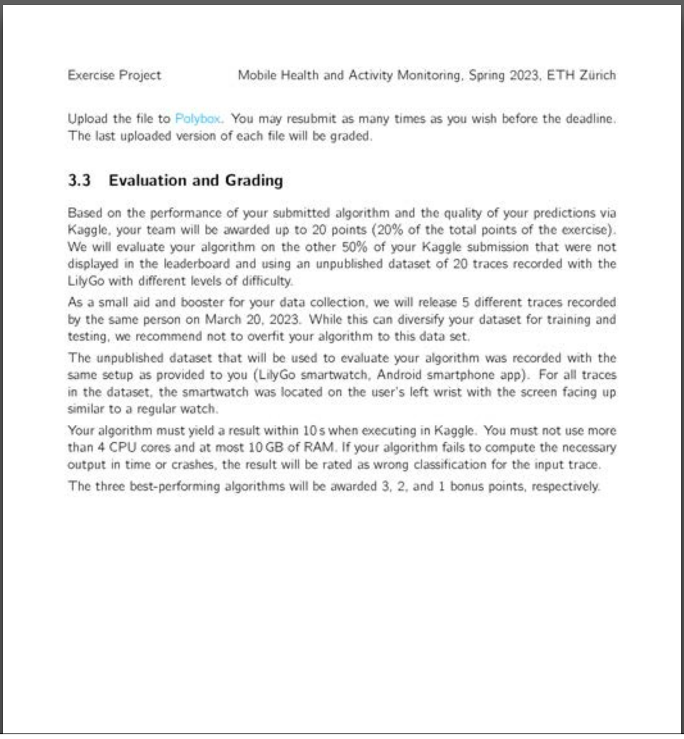
Your algorithm will be evaluated using a dataset recorded by the TA team. Your algorithm must not rely on any signal types other than the data from the LilyGo smartwatch (i.e., accelerometer, gyroscope, magnetometer, IMU temperature sensor). Please note:

- Step counting is a problem for which several algorithms have been proposed. You are free to build on existing algorithms or Python modules, but you may have to optimize them for the best results.
- The following paper may serve as one possible starting point: Walk Detection and Step Counting on Unconstrained Smartphones by Brajdic et al. This paper will also be the first reading assignment of the course. Note that the algorithms they propose are designed for use in a smartphone rather than a smartwatch, but they still offer insight into possible approaches. Google Scholar is useful to find additional related publications on the topic. Starting with Brajdic et al.'s paper, you can find related work in the articles they cited as well as by using 'cited by' in Google Scholar.
- To achieve optimal results, recording your own training data to inspect and understand the LilyGo's signal quality and characteristics will be useful. Make sure to annotate ground truth values for the number of steps you take when recording, which will later facilitate processing. To improve the robustness of your algorithm, it may be helpful for some of your training data to be 'dirty' (e.g. include activity other than walking).
- The definition of a step is described in Subsection 5.1

3.2 Submission

To be evaluated, you'll have to take part in the step count Kaggle competition and submit a Python file via Polybox. The Python file should follow the submission template in our git repository.

- Take part in the step count Kaggle competition. We will upload a set of 20 traces to Kaggle for which you can make predictions but won't see the actual step counts. For the first 10



Upload the file to Polybox. You may resubmit as many times as you wish before the deadline. The last uploaded version of each file will be graded.

3.3 Evaluation and Grading

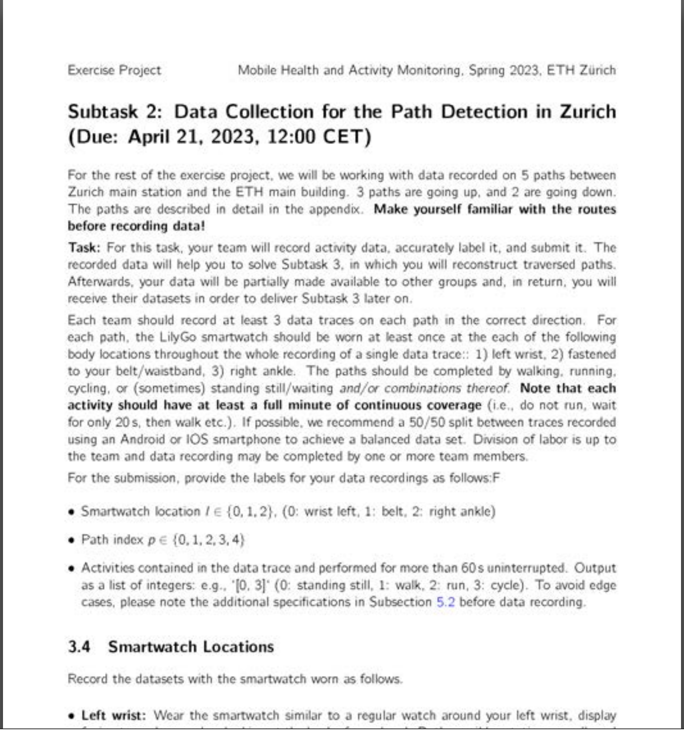
Based on the performance of your submitted algorithm and the quality of your predictions via Kaggle, your team will be awarded up to 20 points (20% of the total points of the exercise). We will evaluate your algorithm on the other 50% of your Kaggle submission that were not displayed in the leaderboard and using an unpublished dataset of 20 traces recorded with the LilyGo with different levels of difficulty.

As a small aid and booster for your data collection, we will release 5 different traces recorded by the same person on March 20, 2023. While this can diversify your dataset for training and testing, we recommend not to overfit your algorithm to this data set.

The unpublished dataset that will be used to evaluate your algorithm was recorded with the same setup as provided to you (LilyGo smartwatch, Android smartphone app). For all traces in the dataset, the smartwatch was located on the user's left wrist with the screen facing up similar to a regular watch.

Your algorithm must yield a result within 10 s when executing in Kaggle. You must not use more than 4 CPU cores and at most 10 GB of RAM. If your algorithm fails to compute the necessary output in time or crashes, the result will be rated as wrong classification for the input trace.

The three best-performing algorithms will be awarded 3, 2, and 1 bonus points, respectively.



Subtask 2: Data Collection for the Path Detection in Zurich (Due: April 21, 2023, 12:00 CET)

For the rest of the exercise project, we will be working with data recorded on 5 paths between Zurich main station and the ETH main building. 3 paths are going up, and 2 are going down. The paths are described in detail in the appendix. **Make yourself familiar with the routes before recording data!**

Task: For this task, your team will record activity data, accurately label it, and submit it. The recorded data will help you to solve Subtask 3, in which you will reconstruct traversed paths. Afterwards, your data will be partially made available to other groups and, in return, you will receive their datasets in order to deliver Subtask 3 later on.

Each team should record at least 3 data traces on each path in the correct direction. For each path, the LilyGo smartwatch should be worn at least once at the each of the following body locations throughout the whole recording of a single data trace:: 1) left wrist, 2) fastened to your belt/waistband, 3) right ankle. The paths should be completed by walking, running, cycling, or (sometimes) standing still/waiting *and/or combinations thereof.* **Note that each activity should have at least a full minute of continuous coverage** (i.e., do not run, wait for only 20 s, then walk etc.). If possible, we recommend a 50/50 split between traces recorded using an Android or IOS smartphone to achieve a balanced data set. Division of labor is up to the team and data recording may be completed by one or more team members.

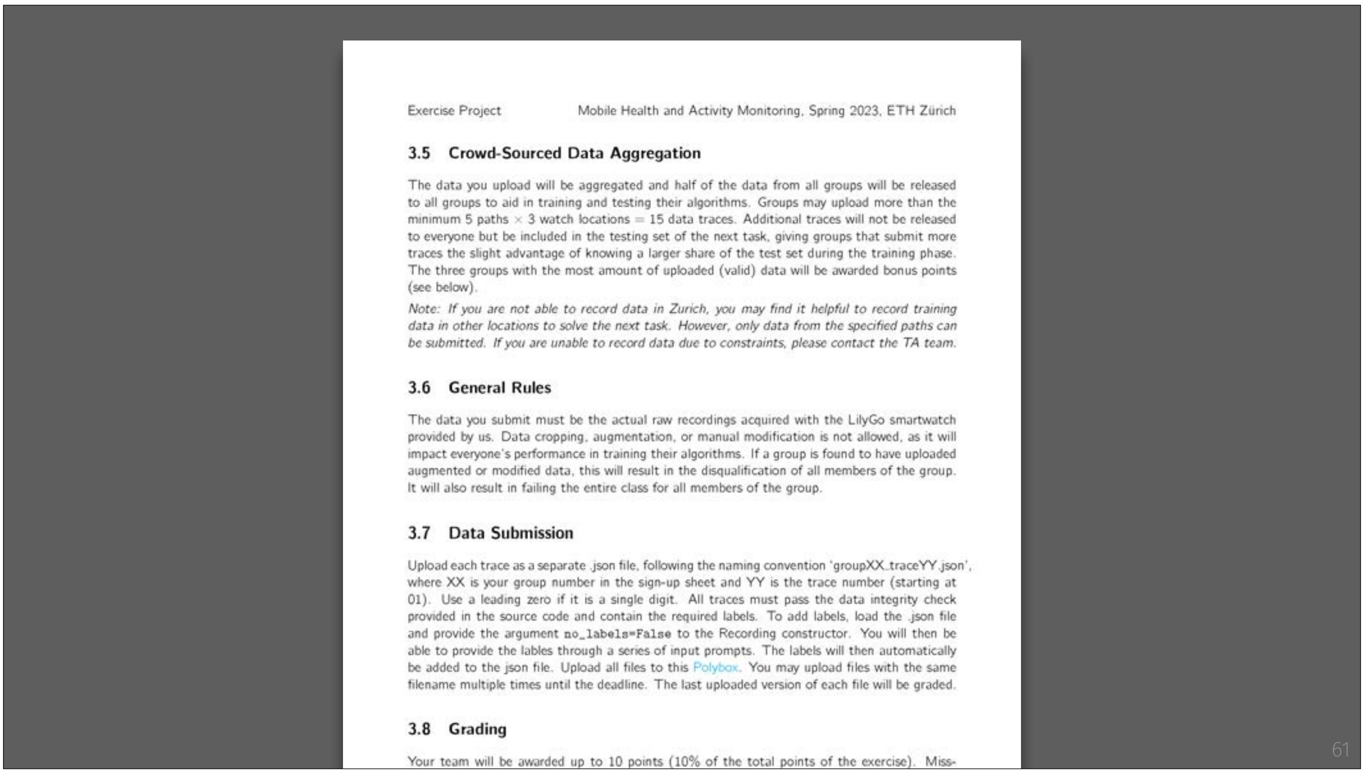
For the submission, provide the labels for your data recordings as follows:F

- Smartwatch location $l \in \{0, 1, 2\}$, (0: wrist left, 1: belt, 2: right ankle)
- Path index $p \in \{0, 1, 2, 3, 4\}$
- Activities contained in the data trace and performed for more than 60 s uninterrupted. Output as a list of integers: e.g., '[0, 3]' (0: standing still, 1: walk, 2: run, 3: cycle). To avoid edge cases, please note the additional specifications in Subsection 5.2 before data recording.

3.4 Smartwatch Locations

Record the datasets with the smartwatch worn as follows.

- **Left wrist:** Wear the smartwatch similar to a regular watch around your left wrist, display



3.5 Crowd-Sourced Data Aggregation

The data you upload will be aggregated and half of the data from all groups will be released to all groups to aid in training and testing their algorithms. Groups may upload more than the minimum 5 paths × 3 watch locations = 15 data traces. Additional traces will not be released to everyone but be included in the testing set of the next task, giving groups that submit more traces the slight advantage of knowing a larger share of the test set during the training phase. The three groups with the most amount of uploaded (valid) data will be awarded bonus points (see below).

Note: If you are not able to record data in Zurich, you may find it helpful to record training data in other locations to solve the next task. However, only data from the specified paths can be submitted. If you are unable to record data due to constraints, please contact the TA team.

3.6 General Rules

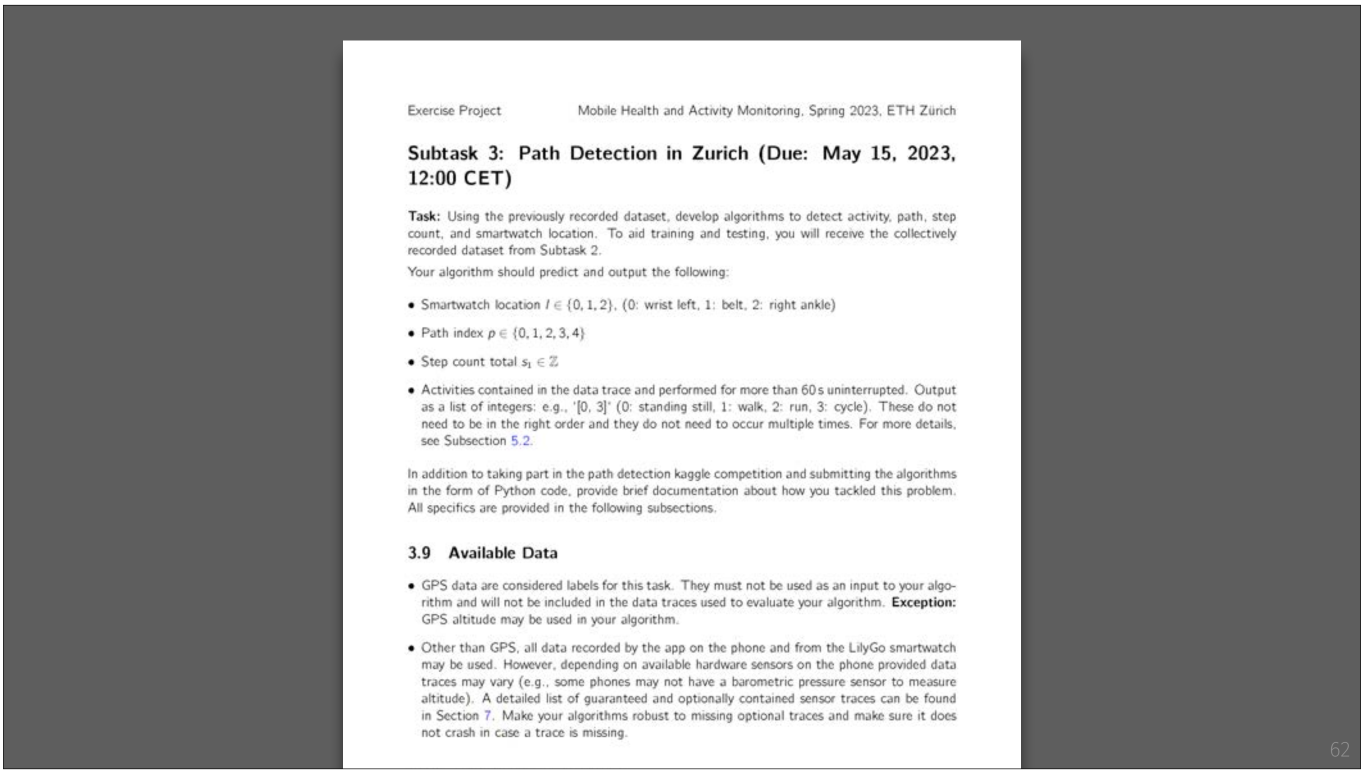
The data you submit must be the actual raw recordings acquired with the LilyGo smartwatch provided by us. Data cropping, augmentation, or manual modification is not allowed, as it will impact everyone's performance in training their algorithms. If a group is found to have uploaded augmented or modified data, this will result in the disqualification of all members of the group. It will also result in failing the entire class for all members of the group.

3.7 Data Submission

Upload each trace as a separate .json file, following the naming convention 'groupXX_traceYY.json', where XX is your group number in the sign-up sheet and YY is the trace number (starting at 01). Use a leading zero if it is a single digit. All traces must pass the data integrity check provided in the source code and contain the required labels. To add labels, load the .json file and provide the argument `no_labels=False` to the Recording constructor. You will then be able to provide the lables through a series of input prompts. The labels will then automatically be added to the json file. Upload all files to this Polybox. You may upload files with the same filename multiple times until the deadline. The last uploaded version of each file will be graded.

3.8 Grading

Your team will be awarded up to 10 points (10% of the total points of the exercise). Miss-



Subtask 3: Path Detection in Zurich (Due: May 15, 2023, 12:00 CET)

Task: Using the previously recorded dataset, develop algorithms to detect activity, path, step count, and smartwatch location. To aid training and testing, you will receive the collectively recorded dataset from Subtask 2.

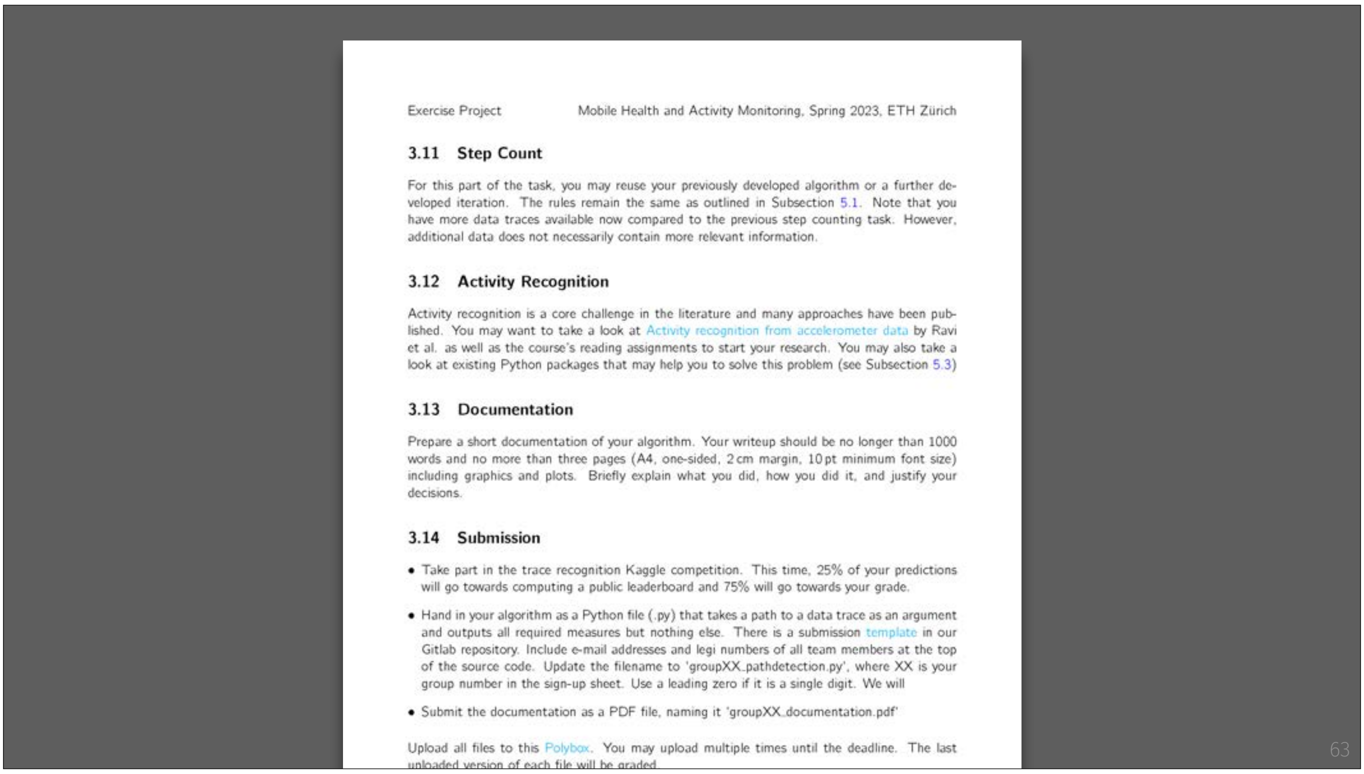
Your algorithm should predict and output the following:

- Smartwatch location $l \in \{0, 1, 2\}$, (0: wrist left, 1: belt, 2: right ankle)
- Path index $p \in \{0, 1, 2, 3, 4\}$
- Step count total $s_1 \in \mathbb{Z}$
- Activities contained in the data trace and performed for more than 60 s uninterrupted. Output as a list of integers: e.g., '[0, 3]' (0: standing still, 1: walk, 2: run, 3: cycle). These do not need to be in the right order and they do not need to occur multiple times. For more details, see Subsection 5.2.

In addition to taking part in the path detection kaggle competition and submitting the algorithms in the form of Python code, provide brief documentation about how you tackled this problem. All specifics are provided in the following subsections.

3.9 Available Data

- GPS data are considered labels for this task. They must not be used as an input to your algorithm and will not be included in the data traces used to evaluate your algorithm. **Exception:** GPS altitude may be used in your algorithm.
- Other than GPS, all data recorded by the app on the phone and from the LilyGo smartwatch may be used. However, depending on available hardware sensors on the phone provided data traces may vary (e.g., some phones may not have a barometric pressure sensor to measure altitude). A detailed list of guaranteed and optionally contained sensor traces can be found in Section 7. Make your algorithms robust to missing optional traces and make sure it does not crash in case a trace is missing.



3.11 Step Count

For this part of the task, you may reuse your previously developed algorithm or a further developed iteration. The rules remain the same as outlined in Subsection 5.1. Note that you have more data traces available now compared to the previous step counting task. However, additional data does not necessarily contain more relevant information.

3.12 Activity Recognition

Activity recognition is a core challenge in the literature and many approaches have been published. You may want to take a look at Activity recognition from accelerometer data by Ravi et al. as well as the course's reading assignments to start your research. You may also take a look at existing Python packages that may help you to solve this problem (see Subsection 5.3)

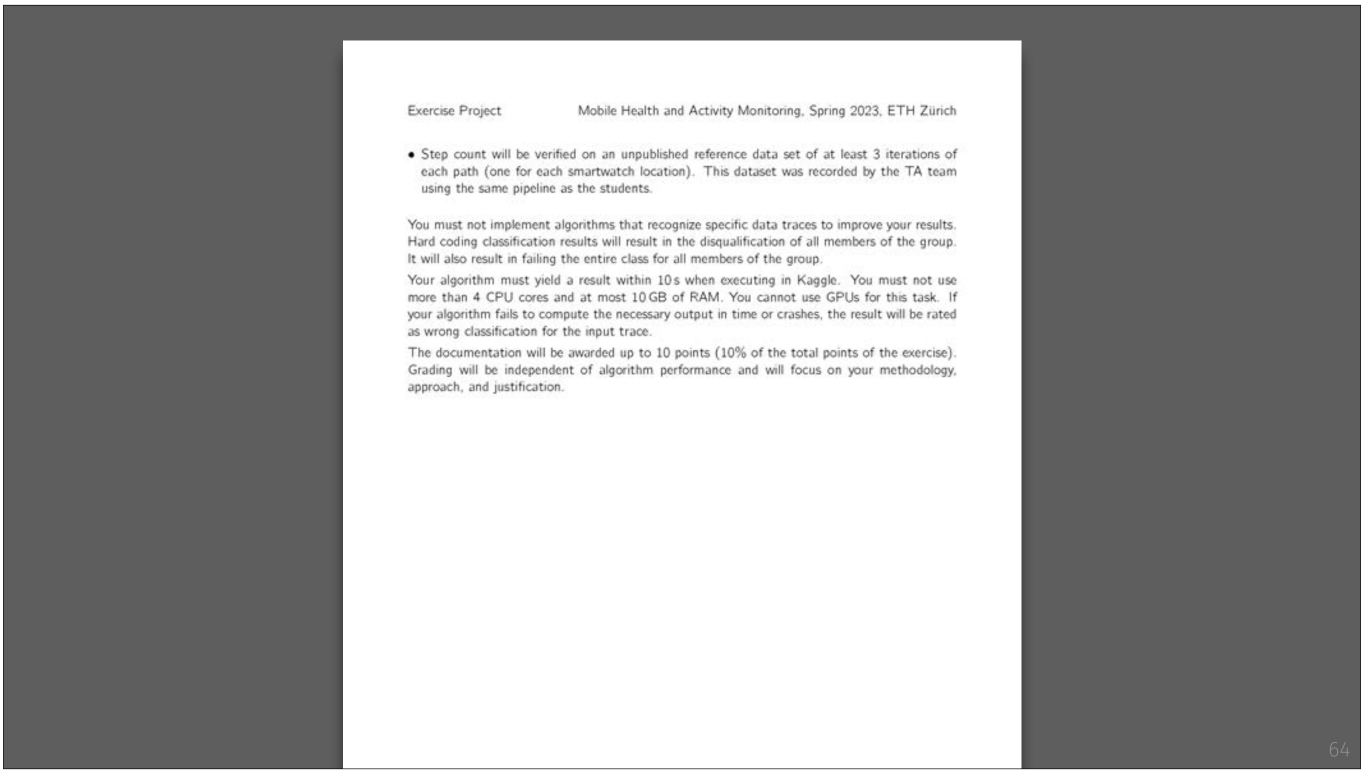
3.13 Documentation

Prepare a short documentation of your algorithm. Your writeup should be no longer than 1000 words and no more than three pages (A4, one-sided, 2 cm margin, 10 pt minimum font size) including graphics and plots. Briefly explain what you did, how you did it, and justify your decisions.

3.14 Submission

- Take part in the trace recognition Kaggle competition. This time, 25% of your predictions will go towards computing a public leaderboard and 75% will go towards your grade.
- Hand in your algorithm as a Python file (.py) that takes a path to a data trace as an argument and outputs all required measures but nothing else. There is a submission template in our Gitlab repository. Include e-mail addresses and legi numbers of all team members at the top of the source code. Update the filename to 'groupXX_pathdetection.py', where XX is your group number in the sign-up sheet. Use a leading zero if it is a single digit. We will
- Submit the documentation as a PDF file, naming it 'groupXX_documentation.pdf'

Upload all files to this Polybox. You may upload multiple times until the deadline. The last uploaded version of each file will be graded.

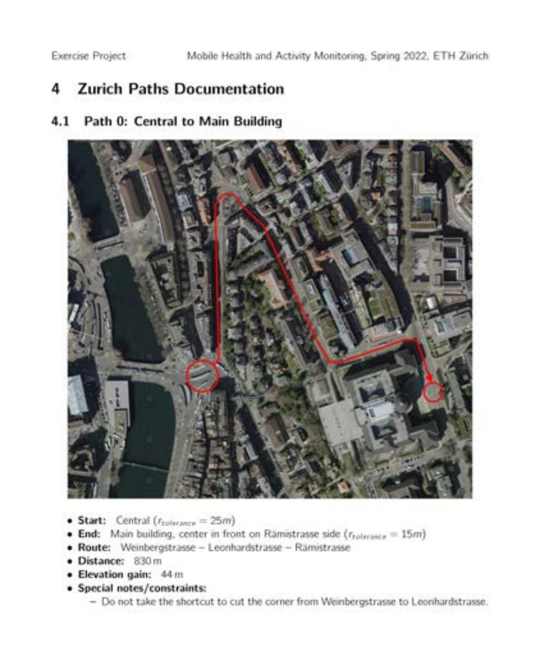


- Step count will be verified on an unpublished reference data set of at least 3 iterations of each path (one for each smartwatch location). This dataset was recorded by the TA team using the same pipeline as the students.

You must not implement algorithms that recognize specific data traces to improve your results. Hard coding classification results will result in the disqualification of all members of the group. It will also result in failing the entire class for all members of the group.

Your algorithm must yield a result within 10 s when executing in Kaggle. You must not use more than 4 CPU cores and at most 10 GB of RAM. You cannot use GPUs for this task. If your algorithm fails to compute the necessary output in time or crashes, the result will be rated as wrong classification for the input trace.

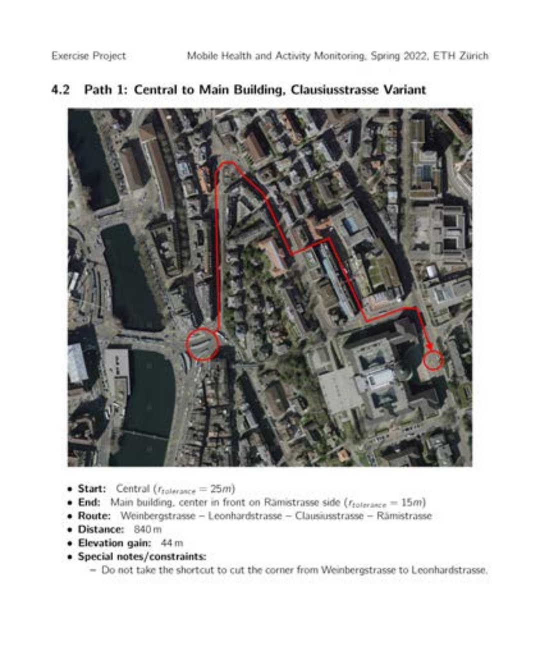
The documentation will be awarded up to 10 points (10% of the total points of the exercise). Grading will be independent of algorithm performance and will focus on your methodology, approach, and justification.



4 Zurich Paths Documentation

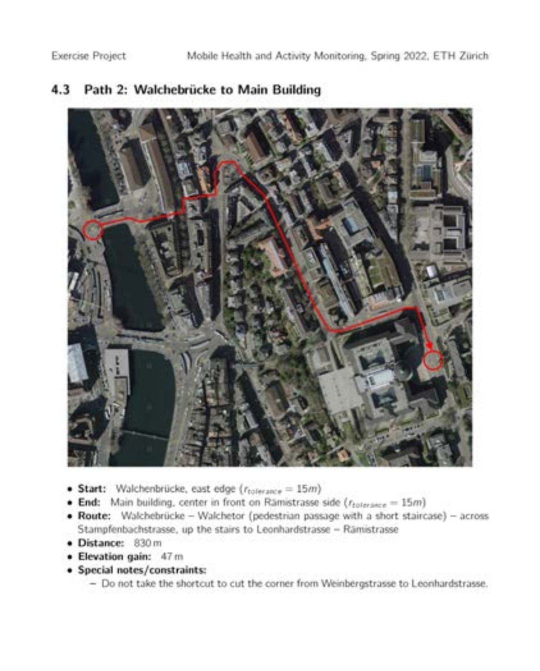
4.1 Path 0: Central to Main Building

- **Start:** Central ($r_{tolerance} = 25m$)
- **End:** Main building, center in front on Rämistrasse side ($r_{tolerance} = 15m$)
- **Route:** Weinbergstrasse – Leonhardstrasse – Rämistrasse
- **Distance:** 830 m
- **Elevation gain:** 44 m
- **Special notes/constraints:**
  - Do not take the shortcut to cut the corner from Weinbergstrasse to Leonhardstrasse.



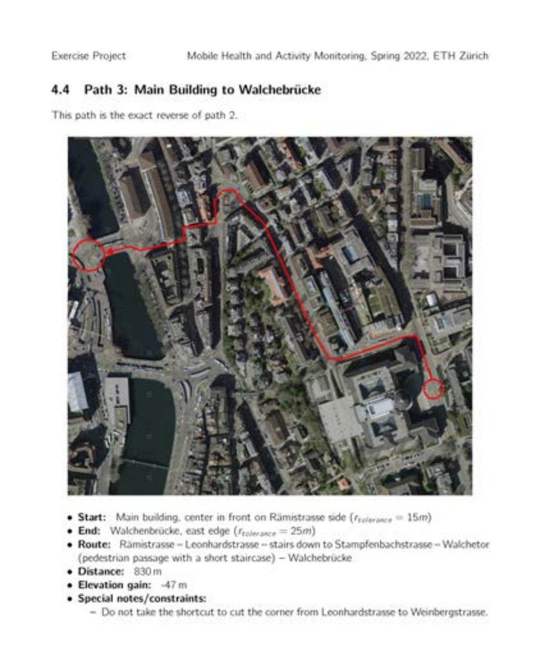
4.2 Path 1: Central to Main Building, Clausiusstrasse Variant

- **Start:** Central ($r_{tolerance} = 25m$)
- **End:** Main building, center in front on Rämistrasse side ($r_{tolerance} = 15m$)
- **Route:** Weinbergstrasse – Leonhardstrasse – Clausiusstrasse – Rämistrasse
- **Distance:** 840 m
- **Elevation gain:** 44 m
- **Special notes/constraints:**
  - Do not take the shortcut to cut the corner from Weinbergstrasse to Leonhardstrasse.



4.3 Path 2: Walchebrücke to Main Building

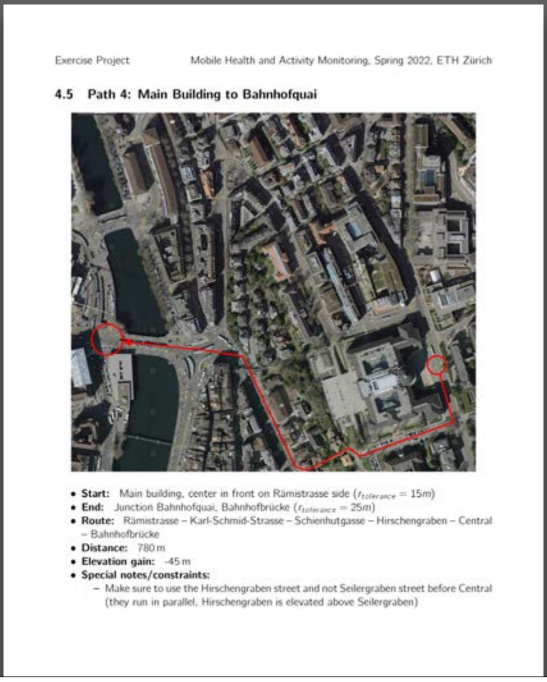
- **Start:** Walchenbrücke, east edge ($r_{tolerance} = 15m$)
- **End:** Main building, center in front on Rämistrasse side ($r_{tolerance} = 15m$)
- **Route:** Walchebrücke – Walchetor (pedestrian passage with a short staircase) – across Stampfenbachstrasse, up the stairs to Leonhardstrasse – Rämistrasse
- **Distance:** 830 m
- **Elevation gain:** 47 m
- **Special notes/constraints:**
  - Do not take the shortcut to cut the corner from Weinbergstrasse to Leonhardstrasse.



4.4 Path 3: Main Building to Walchebrücke

This path is the exact reverse of path 2.

- **Start:** Main building, center in front on Rämistrasse side ($r_{tolerance} = 15m$)
- **End:** Walchenbrücke, east edge ($r_{tolerance} = 25m$)
- **Route:** Rämistrasse – Leonhardstrasse – stairs down to Stampfenbachstrasse – Walchetor (pedestrian passage with a short staircase) – Walchebrücke
- **Distance:** 830 m
- **Elevation gain:** -47 m
- **Special notes/constraints:**
  - Do not take the shortcut to cut the corner from Leonhardstrasse to Weinbergstrasse.



Exercise Project Mobile Health and Activity Monitoring, Spring 2022, ETH Zürich

4.5 Path 4: Main Building to Bahnhofquai

- **Start:** Main building, center in front on Rämistrasse side ($r_{tolerance} = 15m$)
- **End:** Junction Bahnhofquai, Bahnhofbrücke ($r_{tolerance} = 25m$)
- **Route:** Rämistrasse – Karl-Schmid-Strasse – Schienhutgasse – Hirschengraben – Central – Bahnhofbrücke
- **Distance:** 780 m
- **Elevation gain:** -45 m
- **Special notes/constraints:**
  - Make sure to use the Hirschengraben street and not Seilergraben street before Central (they run in parallel. Hirschengraben is elevated above Seilergraben)

plus

tips, tricks, and rules

Python on-boarding code

sensor and data documentation

exercise deadlines

March 6, 2023	create teams
March 13, 2023	system setup (soft)
March 27, 2023	Subtask 1: step counting
April 21, 2023	Subtask 2: data collection for path detection
May 15, 2023	Subtask 3: path detection in Zurich

next time

what is “Ubiquitous computing”?

explicit vs. implicit input

IMUs — Swiss army knife of activity, motion, and context sensing

end

prof. christian holz
mobile health & activity monitoring, spring 2023

sensing,
interaction &
perception lab