Making (Non-)Sense—A Playful and Explorative Approach to Teaching AI Intuition for the Design of Sensor-Based Interactions

Rahel Flechtner
HfG Schwäbisch Gmünd
University of Applied Sciences
rahel.flechtner@hfg-gmuend.de

Jakob Kilian
Köln International School of Design (KISD)
jakob.kilian@th-koeln.de

Abstract
As artificial intelligence (AI) technologies become increasingly important for designing human-computer interactions and user experiences, designers must prepare for the challenge of developing meaningful, creative, and technically feasible AI-based systems. We present a teaching format that we implemented to equip design students with the necessary intuition for AI technologies to develop sensor-based AI-driven interactions. The format consisted of two parts: a role play, which provided a playful, low-threshold introduction to the basics of machine learning for classifying sensor data; and an exploratory part, supported by readymade hardware and software modules, which enabled active engagement with the technology to support creative ideation processes. With this teaching format, we met our teaching objectives of increasing students' technical literacy, teaching the technical language, and providing the necessary tools and knowledge for working with technology as creative material.

CCS Concepts • Human-centered computing – Human computer interaction (HCI).
Keywords: Design Education, Artificial Intelligence
DOI: https://doi.org/10.1145/3587399.3587410

1. Introduction

Artificial intelligence (AI) technologies are becoming increasingly important for the design of human-computer interactions and user experiences. Consequently, the topic must be integrated into design education and the design curriculum [8]. For a meaningful and informed engagement with AI technologies, new challenges arise for designers [20]. These include but are not limited to the following: 1) Designers often lack an understanding of a given AI technology's technical capabilities and restrictions. This makes it difficult for them to imagine feasible AI solutions for a specific problem [20]; 2) The lack of technical knowledge can cause designers to struggle to conceptualize new and creative ways of using AI technologies. As a result, design-led innovations in the field of machine learning (ML) are still rare [4]. 3) The complexity and technical depth of the technology may pose obstacles for designers to prototype their AI-based ideas [20] and to actively engage with the technology in iterative creative processes.

With our teaching format, we approached these challenges and contributed to equipping design students with the necessary intuition that prepares them to create meaningful, creative, and technically feasible AI-based systems and user experiences [8]. Our workshop concept focused on a specific subcategory of AI technology, the processing and classification of sensor data using ML, and its potential for interaction design. ML models are increasingly optimized to consume less energy and require less memory. Hardware capable of running ML models, such as microcontrollers, is also becoming miniaturized, more affordable, and more powerful. This technical progress offers rapidly

growing opportunities for the design and development of new, engaging, and customizable real-time interactions.
In this study, we present our teaching format, which aims to introduce students to the basics of ML and sensor data in a playful way and enable active exploration of the technology and its potential for the development of novel AI-based interactions. The format involved two parts: 1) a role play that allows low-barrier access to understanding the technical capabilities and restrictions of the technology and 2) hands-on engagement with the technology using readymade sensing and acting modules. We describe our approach in detail and reflect on the implementation of the activities. We also provide teaching materials and access to the code.

2. Related Work

Designers are experienced in transforming known technologies into novel and valuable products or services [19]. Regarding new technologies, part of the research in the Human-Computer-Interaction (HCI) community focuses on making a particular technology or medium more accessible to designers. This includes approaches that frame emerging technologies as design material, such as „software as design material" [15], „electronics as material" [2], „machine learning as design material" [4], „artificial intelligence as design material" [11], and „data as design material" [13]. The characterization of a technology as a design material implies design-driven approaches to engaging with the technology, such as the act of „reflection in action" [16], in which the ideation process occurs while working with the material. Engaging in reflective processes during the hands-on interaction with the material aims to facilitate the development of new, innovative applications. However, according to Yang [19], this process requires designers to have a tacit understanding of the materials they are working with. With new technology, this creative process happens less easily because designers lack at-hand knowledge of the material's capabilities. Therefore, designers must become familiar with emerging technologies to innovate [19]. Not being able to implement and test an idea significantly limits reflection in action. This also applies to technologies, especially those as complex as aspects of AI. As Dove et al. [4] summarize, „ML is a more difficult design material to work with."

Recently, the HCI community has been increasingly researching how to facilitate the design and innovation of human-AI interactions [20]. Among others, the following three approaches are discussed: 1) The technical literacy of designers must be improved to understand AI technologies' capabilities [8,20]. This also includes the development of a precise language for communicating about AI topics in order to develop a mutual understanding in the increasing collaboration among designers, data scientists, and AI experts [9]. 2) Designers must be able to engage with the technology in a hands-on way to support ideation [6,17,20]. 3) As data plays an essential role in understanding and developing AI-based interactions and applications, it is crucial for designers working with and designing AI technologies to acquire the necessary data literacy[19] and engage with data in a meaningful way during the design process [13].

In addition to textbooks, several resources are available online that can support designers in developing the necessary technical literacy in the context of AI, such as online courses and resource collections [1,7,21–23]. However, these sources are usually very extensive, require much time to work on, and are often difficult to integrate into teaching formats. No-code tools such as Teachable Machine [24], Tiny Motion Trainer [25], and Wekinator [5] facilitate easily accessible, hands-on engagement with AI technology. In electronics and physical computing, physical toolkits such as Blokdots [3] make hardware components readily available to designers, allowing them to quickly and efficiently implement and test ideas in prototypes. All these tools often work by reducing the complexity of a technology to make it more accessible in terms of technical knowledge and time. This makes them well suited for use in teaching per se. However, they are often limited in their capabilities for implementing complex ideas. Professional tools such as the Edge Impulse [26] software platform offer extensive capabilities for realizing and testing ideas but require a certain amount of prior knowledge.

Concerning AI in design education, few examples can be found of providing the right amount of technical knowledge and enabling sufficient hands-on engagement with the technology in a time-limited setting such as a workshop. One example that does not explicitly focus on AI technologies but on the use and potential of sensor data in the design process is the teaching format by Lallemand et al. [14]. They propose a practice-based teaching activity that introduces interaction design students to the potential of sensors for user research and ideation without the need for technical knowledge.

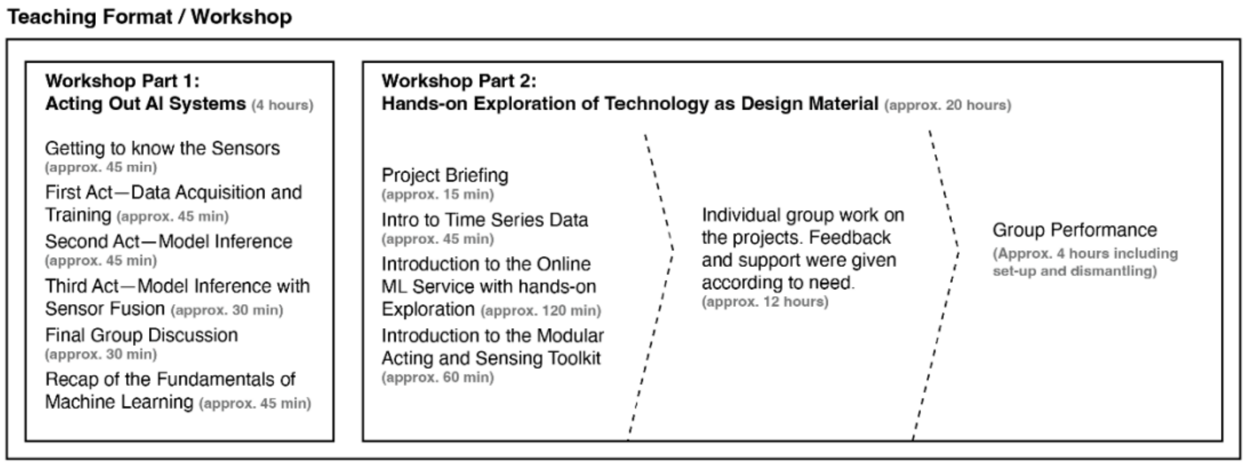

Figure 1: Overview of the workshop schedule, including the two parts and the associated activities.

In summary, designers must be prepared for new challenges in a design-led approach to developing novel and meaningful AI-based systems. Several resources and tools exist that can be used in teaching. However, there is still a lack of examples and reports on how implementation in design education can work in practice. We aim to contribute to this with our teaching format.

3. Teaching Format

Based on the challenges that designers face when using AI technology in their design processes, our teaching format addressed the following learning objectives: 1) It should expand students' technical knowledge for them to understand the capabilities and limitations of the given AI technology—in our specific case, the processing and classification of sensor data. As data is crucial for developing AI-based interactions and applications, students should also develop the necessary data competency. 2) Students should be introduced to the relevant technical vocabulary to enable them to work in interdisciplinary teams with data scientists and AI experts. 3) It should allow them to actively engage with the technology to enter into creative and reflective ideation processes that facilitate the development of new, innovative applications. This includes providing them the necessary skills to implement and test their ideas as prototypes.

We held the teaching format (referred to as a „workshop") twice, with different groups of students: once at the HfG Schwäbisch Gmünd and once at the Köln International School of Design (KISD). In the first run, we planned the workshop for three consecutive days. This duration proved too short, so we scheduled the second implementation for four days. Our description of the format refers to the second implementation.

In the following, we provide a short summary of the procedure. In the subsequent sections (3.1-3.3), we describe the individual parts and components in detail and provide in-depth information on the process, organization, and technical implementation.

Our teaching format was divided into two parts (fig 1): For low-barrier access to the topic, the first part emphasized a playful introduction to the technical capabilities and limitations of the technology without requiring technical skills. The students engaged in a role play in which they embodied the different components of a sensor-based AI system to gain a basic understanding of the technology. This part of our teaching format was inspired by the approach of Lallemand et al. [14]. Initiated by the role play, we delved into the basics of ML and examined the fundamentals of neural networks and their functional principles. The first part of the workshop lasted about four hours.

The goal of the second workshop part was to actively engage with the technology as a design material to explore its potential for new forms of interaction beyond simply extrapolating known use cases. To this end, the students received a project briefing focused on exploratory rather than problem-solving approaches. They were introduced to a tool that enables the code-free training of ML models and their deployment on microcontrollers. At the same time, the second part of the workshop addressed key questions about training a model with custom data, the steps necessary for data collection and preparation, crucial parameters in the training process, and strategies for model improvement. The goal of this second part was to run an individually trained model on a microcontroller and present the developed interactions in a group performance. This part of the workshop was held over three and a half days. The material, code files, and a short video documentation of the workshop can be found in the project repository: https://gitlab.rlp.net/kitegg/public/making-nonsense.

3.1. Workshop Part 1: Acting Out AI Systems

In the first part of the workshop, students role-played the components of a sensor-based AI system. The aim was to develop an understanding of sensor data and the basic principles of data collection, processing, and interpretation by ML systems. To this end, the students playfully explored which types of data can be captured by the given sensors, how this data represents their environment, and how this information can be used to conclude specific events. For the role play, the group was divided into subgroups with different roles: two actors, two observers, and several groups of „intelligent systems," each consisting of a „human sensor" and two people representing the algorithm (fig. 2).

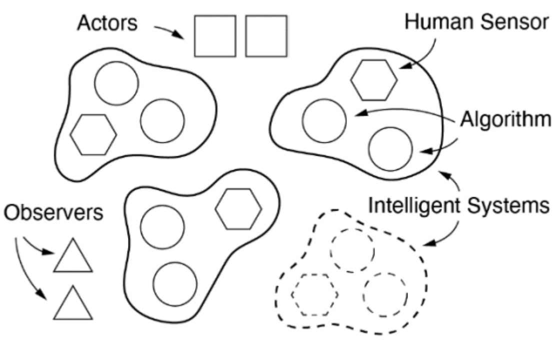

Figure 2: Groupes and different roles assigned to the students for the role play.

Each group picked a sensor from a predefined group of sensors. The following sensors were available: a photoresistor, an accelerometer, a microphone, a thermal camera, and an infrared distance sensor. The groups familiarized themselves with the given sensors by researching their functionality online. The objective was to determine how the sensor technically works, what it measures, and what data it outputs. The students compiled this information into a sensor profile and presented their findings to the group.

3.1.1. First Act—Data Acquisition and Training Phase

The two actors were then asked to play a scene. The aim was to reenact a simple, everyday situation in which sensor data can be used to conclude certain events. In this case, the instructors chose an office situation in which the students had to analyze whether a person was working. One person was asked to sit on a chair in front of a screen and operate a mouse and keyboard. The other actor played a colleague who would later enter the room. A different situation could have been chosen; this point is further addressed in the discussion. In the first act, the scene was performed by only one person. The actor was briefed in advance to perform obvious gestures such as typing on the keyboard, getting up from the chair, or leaving the room.

Data Acquisition: The groups of „intelligent systems" observed the scene and were asked to consider what information could be derived from the data of their specific sensor in order to conclude whether a person was working. The placement of the imaginary sensor in the room could be chosen freely. However, only one sensor could be used, and its position had to be maintained until the end of the role-playing activity. The students were then asked to formalize their decisions on the „sensor mission sheet" by answering questions such as where precisely the sensor should be placed, what exactly it should record, in what format, and when the data should be sent. For this task, it was important that the sensors only convey information in the form of data, not yet an interpretation of this data. On one hand, the sensor mission sheets aimed to reflect the significance of specific data and the relevance of well-thought-out data acquisition. On the other, it guided the human sensors in their later data collection process.

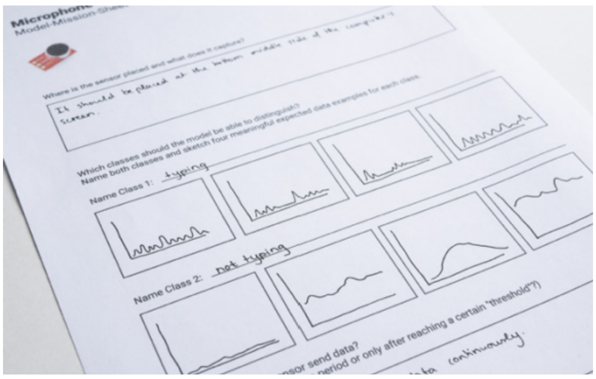

Figure 3: The model mission sheet with data examples for two classes, "typing" and "not typing."

Training Phase: After defining the data acquisition process, the groups were asked to think about the model side of their system. They defined how they could interpret the data received from the sensors and which conclusions the algorithm could draw about the scene. To this end, they completed the model mission sheet (fig. 3), on which they defined two classes that the model should be able to distinguish by giving them descriptive names and outlining four meaningful expected data examples for each class. By completing the sheet, the

Figure 5: ”Human sensors” recording data in various data formats such as graphs, numbers, and color-coded temperature ranges.

individual groups conducted something similar to the training process of an ML model: they agreed on which patterns they wanted to recognize in their data and which class name or event to assign to each pattern.

3.1.2. Second Act—Model Inference
The role play then progressed to the second act, model inference. This part required each group to apply its model to the live data from the human sensors and to make real-time predictions about the events occurring in the scene. This involved a technical setup, which the students realized together (fig. 4): The human sensors remained in the room with the actors and observers. The people representing the algorithm moved to another room. The human sensors were then each given a roll of paper on which they were to continuously document their data stream. Using a webcam and a video conferencing tool, a live data stream video was transmitted to the respective algorithm team, which received this stream in the next room. Using a group chat as a communication platform, they communicated their predictions (i.e., the conclusions drawn from the data they received about possible events in the scene) by posting the respective class names when recognized. The observers in the main room should follow the prediction stream in the group chat and compare it with the events happening in the scene. This allows them to later give feedback to the teams on correct and incorrect predictions.

To prevent the groups from overly familiarizing themselves with the data and the expected events in advance, the sensor and mission sheets were swapped among the groups (i.e., each group was assigned a sensor or model mission they had not planned themselves). Parallels were made with overfitting, in which a model adapts too strongly to the training data and can no longer correctly interpret unknown data. In a second briefing, the actors were instructed to play a slightly more complex scene and to incorporate actions that the teams had not yet seen in the training run-through (e.g., the second person should now enter the room through the door, or the two people should interact with each other).

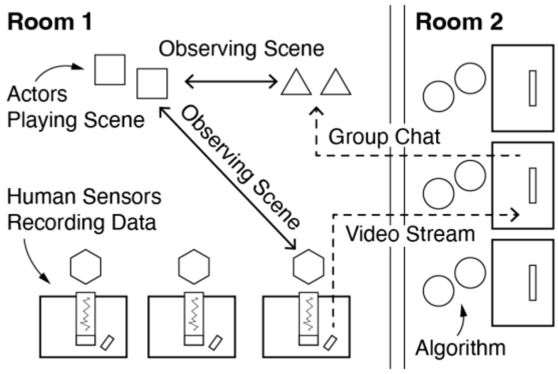

Figure 4: Technical setup and procedure of the role play in the second act.

After the technical setup, the starting signal was given for the second act: The actors played the scene, the human sensors recorded their data (fig. 5), and the data stream was transmitted via video to their algorithm team in the next room. This team interpreted the data and sent its predictions to the observer team via group chat. The latter compared the predictions with the reality of the scene being played.

3.1.3. Third Act—Model Inference with Sensor Fusion
After the second act, the participants reconvened. The algorithm groups reported on their experiences, including the quality of the sensor data they had received, what they could deduce from the data, where they needed clarification, or what had not worked so well. The observers also shared their experiences about which events were reliably identified, mislabeled, or not recognized. The findings

from the discussion should help the groups improve their predictions in the next step; in other words, it serves as a further training phase. The systems were also allowed to recalibrate their human sensors by adjusting the transmission rate or threshold values. An insight that the students gained from this task is that it is often not possible to draw accurate conclusions about certain events with simple sensor data (e.g., the absence of keystrokes could mean that a person is not working, but they could also be reading something on the screen; opening the door could mean one person is leaving the room or that a second person is entering).
Based on these learnings, the concept of sensor fusion was introduced in the third act: The actors performed a similar scene as before, and the individual groups used their improved systems to predict the events happening in the scene. The difference from the second act was that the algorithm groups could now exchange information while making predictions. This allowed information from different sensors to be combined and much more precise predictions to be made based on the merged data. The bundled predictions were again forwarded to the observers via group chat.

3.1.4. Final Group Discussion—Recapitulating the Learnings

After the final act, all participants met again. We discussed what had been learned from the experience and put it into the context of ML. For instance, by realizing that it was possible to make much more precise predictions by bundling different sensor information, the concept of sensor fusion was explained. In addition, some students stated that gaining meaningful information from their sensor data was more difficult than they had thought. At this point, the relevance of a strategic and technically well-thought-out approach for the development and realization of AI-related use cases was discussed with the students.

Another issue raised was the high latency between an event happening in the scene and the prediction being received via group chat, which made it particularly difficult for the observers to evaluate the performance of the models. At this point, the implications of latency and inference speed were discussed. One group reported it could not interpret its sensor data in the first act as it was sent in a completely different format than that noted on the mission sheet. Even though the group members comprehended the data with their human understanding, they precisely followed the instructions on the mission sheet and had to accept the run as a failure. Through this experience, the students learned about potential sources of error in the training or inference process.
The first part of the workshop concluded with a short lecture covering the basics of ML, in which the fundamentals of neural networks and their operating principles were discussed. These included topics such as the structure of a neural network consisting of input, hidden, and output layers; different types of data as input (e.g., pixel data for images or time series data); the significance of labeled data in supervised learning; the concept of classification and the basics of training; and introducing concepts such as weights, backpropagation, and training cycles. Parallels to the role play were emphasized at the relevant points.

3.2. Workshop Part 2: Hands-on Exploration of Technology as Design Material

The second part of the workshop focused on a playful exploration of the technology and its potential for designing new forms of AI-based interactions. To this end, the students were provided with the necessary hardware and software skills to train an ML model with their data. This involved acquiring, preparing, and processing sensor data; training and improving the model; and deploying the model on a microcontroller.

In this part of the workshop, we entered the field of tiny machine learning (TinyML), that is, executing ML model inference directly on microcontrollers and thus performing on-device analyses for different sensing modalities [12]. The advantages of such local data processing are low power consumption (often battery-operated), increased data privacy and security (as the raw data is kept locally), low latency, and increased reliability (due to the independence of networks, remote servers, or services). The complexity and technical depth of the technology presented particular challenges for implementing this part of the workshop. Specifically, the design of the course and the provided material were influenced by the following factors:

- Participants often have little or no prior knowledge of programming, ML, and electronics.
- Despite the use of software tools, higher-level programming languages, and frameworks, the ML training process remains complex and technically demanding, especially the subsequent deployment of the model.
- A wide range of data types (e.g., image, audio, movement), sensor types (digital, analog, single/multidimensional), processors, and boards are available.

- Participants use their own computers to compile the code and transfer it to the microcontrollers. As this involves a variety of models and operating systems, there is a high potential for incompatibilities and errors.
- To not hinder the creative process, participants should be able to move freely without being tied to a power source.
- To speed the development process, examining sensor data and testing the model‘s performance on the microcontroller in real time should be possible. For that, a display is required.

3.2.1. Project Briefing

To leave enough freedom for exploration and experimentation and provide an overarching project briefing, the goal of the second part of the workshop was inspired by the famous yet usually completely analog Rube Goldberg machine [10,18]. The aim was to create a chain reaction of TinyML-based classifiers that detect an input, such as specific properties or patterns of change in the physical environment, and, on detecting the pattern, trigger an output (actuator) that initiates the next element in the chain and so on. By linking readymade sensing and acting modules, the students should create a group-spanning chain reaction to be demonstrated in a performance at the end of the workshop.

Given the short duration of the workshop and the technical challenges, we decided to use an online service for data acquisition, model training, and deployment and to develop a modular toolkit of readymade sensing and acting modules. Using a simple, consistent, and visible interface among these modules (a red and a green LED triggered by the sensing module and a light sensor read by the acting module), we ensured the individual projects could be linked together as required. This allowed participants to focus on designing their individual systems by iteratively exploring the given sensor, its data, its potential for creating novel forms of interaction, and the resulting model.

3.2.2. Online ML Service

We used Edge Impulse (EI) [26], a software-as-a-service tool that guides users through the ML process in a browser without the need for specialized hardware or software to be installed on their machines. It radically simplifies collecting and editing data, organizing and labeling the data, building a model architecture, and performing the actual training process in a visual and guided way (fig. 6). It also offers a certain degree of access to the hyperparameters and reveals the essential steps in the training process. Other available tools hide these steps from the user in favor of simplification, thus offering fewer opportunities for individual control over the process. Furthermore, EI provides an optimized TinyML TensorFlow-lite [27] model as an easy-to-install Arduino library. The EI data forwarder was used to transfer the data to the server via a computer‘s serial port.

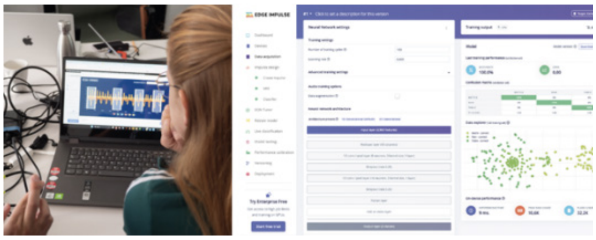

Figure 6: The visual and guided process of data acquisition (left) and model training (right) using the EI online service.

3.2.3. Introducing the Modular Acting and Sensing Toolkit

The toolkit we prepared for the workshop contains two types of modules: the sensing module and the acting module, as shown in figure 7. Both modules consist of a) the microcontroller board, b) a sensor or an actor, c) a power bank for power supply for several hours, d) a prototyping board that allows connecting both types of modules, and e) the interface with a red and green LED on the sensing side using light to transmit the inference result to the photoresistors on the acting side.

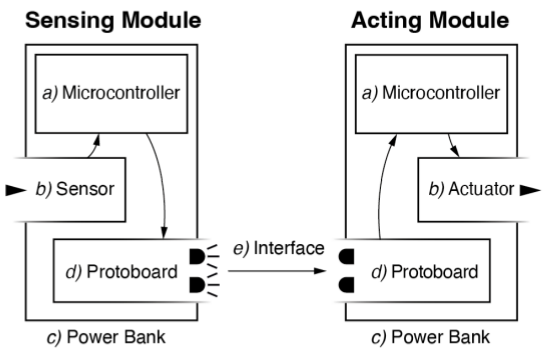

Figure 7: Schematic representation of the sensing and acting modules, which communicate with each other via a simple LED-based interface.

The core component of each module is an M5-StickC PLUS [28] microcontroller based on an ESP32 chip. It is relatively inexpensive, comes with some sensors on board, has an ecosystem of external sensors/actuators including libraries, a small battery for quick experiments, and a display, as shown in figure 8. We provided prewritten

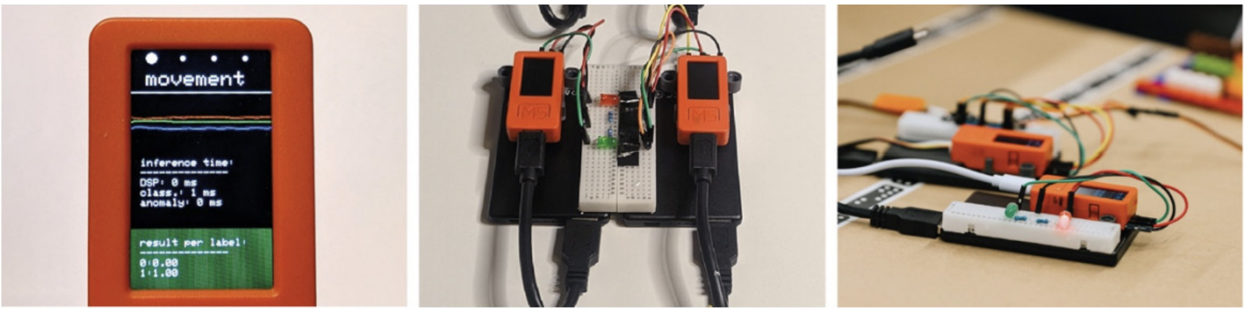

Figure 8: The M5-StickC PLUS microcontroller (left) and the readymade acting and sensing modules (center and right).

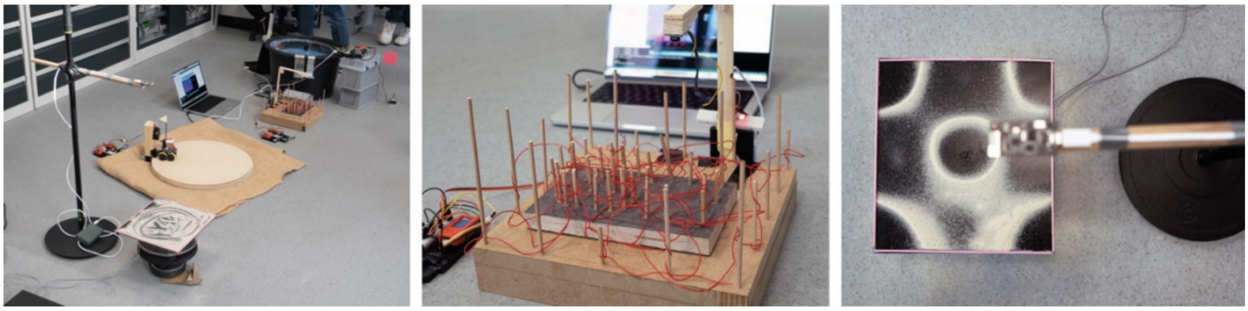

Figure 9: The presentation-ready Rube Goldberg machine performance (left) with exponents detecting hidden temperature patterns in a bunch of cables (center) or visual patterns of audio frequencies (right).

code to stream the sensor data via a serial port to a computer and from there to EI, as well as to run the resulting model and display some information on the screen. Hence, the participants only had to change the parameters for their projects.

In the workshop, the Arduino IDE [29] was used for editing, compiling, and flashing code for the modules. It is easy to install and navigate, supports many boards, and provides all the necessary libraries, board information, and their interfaces for external hardware directly or via plug-ins. In contrast to the programming of more complex systems such as minicomputers, it is deliberately designed for beginners. However, due to the variety of operating systems and USB hardware, unexpected problems and errors may occur that cannot be solved spontaneously during the workshop. We proactively addressed this problem by providing participants with a specially prepared Raspberry Pi system (with all necessary dependencies) to be used as a compile and upload tool in case of errors with their hardware. With networking capabilities and responsive remote access, the system can be used from any laptop, regardless of the operating system or specifications. It allows custom models downloaded from EI to be integrated into code, compiled, and uploaded to the microcontroller of the respective module. This approach was a reliable fallback and ensured that technical obstacles did not impede the workshop's progress. A system that enables remote editing and compilation and subsequent over-the-air (OTA) upload to microcontrollers via a local network or even the internet would be even more preferable.

3.2.4. Procedure

Before starting the hands-on activities of the second workshop part, a brief introduction was given on the basics and terminology of TinyML, microcontrollers as hardware, and the Arduino C++ code as software to be used. The objectives of the second part of the workshop included: 1) exploring the data of a given sensor and using it in the Edge Impulse workflow, 2) designing interesting interactions and training the corresponding classification model, and 3) Refining the model and creating an interactive physical artifact to be integrated into the group performance.

The students worked in groups of two, with each group using a different sensing module, including an inertial measurement unit (IMU), which combines acceleration, gyroscope, and magnetometer data; a microphone; a thermal imaging camera; an infrared distance sensor; and a camera. Regarding the acting modules, the students had to be creative and develop individual solutions. The acting boards were primarily equipped with basic components such as radio modules for controlling radio-controlled sockets, relays, or servo motors for simple movements.

The result was a largely functional Rube Goldberg machine, as seen in figure 9 (left), and in a video on the project repository https://gitlab.rlp.net/kitegg/public/making-nonsense. Further documentation material and the code written for the individual modules can also be found there. Innovative forms of interaction created by the students were, for example, the detection of visual patterns of audio frequencies in vibrating sand using a camera module (fig. 9, right) and the detection of hidden patterns in a bunch of cables based on temperature change that can only be detected by a thermal camera (fig. 9, center).

3.3. Workshop Evaluation

During the second implementation of the teaching format, we conducted an evaluation, which was completed by five of the eight participants. In the field of skills acquisition, four people fully agreed with the statement that their knowledge of AI had been expanded as a result of the course. One person slightly agreed. Four people also fully agreed with the statement that the format had improved their basic understanding of AI. One person slightly agreed with the statement. Four people stated that the tasks set in the course were appropriate, while one person found the tasks rather difficult.
Regarding personal development, students were asked whether they felt more confident in dealing with the topic of AI after the course. One person fully agreed, three slightly agreed, and one answered neutrally. Four people fully agreed that they had gained new ideas and inspiration from the course, while one slightly agreed.
Due to the small number of participants, the evaluation can only provide an initial impression of the student's perception of the format. If the format is carried out again, a larger evaluation with more participants should be sought.

4. Discussion

In line with related literature [6,8,9,13,17,20], and the goal of equipping the students with the necessary intuition for AI technologies, our teaching format pursued the following learning objectives: 1) to increase technical literacy by equipping students with the necessary intuition for AI technologies, which includes the understanding of data; 2) teaching the relevant technical language to enable students to work in cross-disciplinary teams with AI experts and data scientists; and 3) providing them with tools and knowledge to engage with the technology as a design material to support ideation and enable prototyping and testing ideas.

The first part of the workshop focused on a playful and low-threshold introduction to the technical capabilities and limitations of the technology, thus contributing to increasing technical literacy and conveying precise language. We observed that the role-playing method led to the students' very engaged involvement. Without being asked to, some students even presented the results of their research into the functionality of their sensors in the form of a role play. We uncovered and addressed many ML-relevant topics during the role play, including different data types; pitfalls in data acquisition, processing, and interpretation; latency; and sensor fusion. In the short technical lecture following the role play, the students were engaged and able to ask specific questions based on their prior learning. The evaluation results underline that the students were able to expand their (technical) AI knowledge through the workshop. The chosen use case (determining whether a person is working) worked well from a technical point of view but is questionable from a moral perspective. We hope that others will develop more suitable use cases and share them. The role of the observers could also be improved. In some cases, the observers were relatively uninvolved in what was happening, and the findings from the observations were limited. It would perhaps be better not to assign the role permanently but to have one person from the algorithm teams perform the task in rotation. Alternatively, the observers' task could be expanded and thus made more attractive.

The second part of the workshop focused on actively engaging with technology as a design material to facilitate creative processes. The explorative approach should also promote knowledge about the material's capabilities and, thus, technical literacy. In addition, it should impart data expertise and expand the technical language. During the workshop implementation, we observed that the students developed the necessary understanding of data and (Tiny)ML and learned the necessary terminology to handle the EI ML process. Despite the limited time, they managed to train and deploy their models and embed them in an essentially functioning and presentation-ready Rube Goldberg machine performance. The resulting interactions were creative and explorative, far beyond extrapolating known use cases. The high time expenditure and the nonlinearity of the training process were vividly conveyed in practice. Although this led to some frustration for individual students, embracing complexity is a critical learning process for developing AI-based systems. This point also underlines a relevant learning in the context of AI intuition: being able to evaluate the computational costs and their appropriateness in relation to a given design goal to make a conscious decision for or against AI technologies [8]. In this context, one student commented: „Many of our experimental setups could have been realized without AI. Although I thought a lot about where I could effectively integrate AI into a simple experiment, in the end, we built a setup that could theoretically be realized without it."

Using our modular sensing and acting toolkit and the readymade code drastically simplified the technical complexity of the workshop setup. It enabled the students to engage hands-on with the technology with little prior hardware and code knowledge. This aspect was confirmed by student feedback, as one person stated: „I liked the fact that the material was prepared ‚just enough' so that I could figure out the rest myself and save time for other things that would have taken me too long." The evaluation underlines that the students were able to gain new ideas and inspiration regarding AI by engaging with the technology.

Nevertheless, we faced some challenges in the technical implementation: much time was spent installing software and debugging, and there was downtime at EI during the workshop. Using a free online service for training also slowed the ability to iterate fast due to queuing times when the server was under a heavy load. Furthermore, a disadvantage of the readymade modules is that participants have difficulty repeating what they have learned if they do not have the modules.

We conducted the workshop in a one-week consecutive format. However, our proposal constitutes a teaching concept rather than a specific course unit. The two parts of the workshop can easily be separated in time and implemented in other teaching formats (e.g., consecutively over several weeks). The first part of the workshop (section 3.1: Acting out AI Systems) offers a playful and relatively concise (approx. 4 hours) introduction to the technical capabilities and limitations of the technology. Therefore, this first workshop part can be used as a kick-off for studio courses and semester-long projects, whereby the second part of the workshop (section 3.2: Hands-on Exploration of Technology as Design Material) can be modified by providing a project-related briefing. The technical components and the code can be easily adapted for this purpose.

Given a short time frame (e.g., only one day), the workshop could be conducted with a more technical focus. It could be kicked off with the „acting out AI systems" part in the morning, while in the afternoon an introduction to the online ML service could be given including space for hands-on exploration. Within this time, participants could train a simple model for a simple use case (e.g., code word recognition or recognizing a specific gesture). In this case, however, exploring the technology as a design material would fall short.

5. Conclusion

With our teaching format, we contribute to the goal of equipping design students with the necessary intuition for AI technologies that would prepare them to create meaningful, creative, and technically feasible AI-based systems and user experiences [8]. Our format aimed to 1) create a playful, low-threshold introduction to the basics of ML for the classification of sensor data and 2) enable students to explore the technology as creative material through hands-on engagement.

Reflecting on our approach, we conclude that our teaching format has met our overarching learning objectives. Nevertheless, it leaves room for improvement. We encourage other educators to apply our approach, adapt it to their requirements, and develop it further.

ACKNOWLEDGMENTS

This teaching format was developed at the AI+Design Lab at HfG Schwäbisch Gmünd University of Applied Sciences and the Living Objects Lab at the Köln International School of Design (KISD) in Cologne. Both labs are part of the KITeGG research project, funded by the Federal Ministry of Education and Research under the funding code 16DHBKI067. The research project aims to integrate AI into the design curriculum.

REFERENCES

[1] Alexander Amini, Ava Amini, and Sadhana Lolla. Introduction to Deep Learning. Retrieved February 23, 2024 from http://introtodeeplearning.com/

[2] Ayah Bdeir and Paul Rothman. 2012. Electronics as material: littleBits. In Proceedings of the Sixth International Conference on Tangible,

Embedded and Embodied Interaction, 371–374. https://doi.org/10.1145/2148131.2148220
[3] Brücker, Oliver, Labacher, Christoph, and Paeschke, Marcus. Blokdots. Retrieved from https://blokdots.com/
[4] Graham Dove, Kim Halskov, Jodi Forlizzi, and John Zimmerman. 2017. UX Design Innovation: Challenges for Working with Machine Learning as a Design Material. In Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems, 278–288. https://doi.org/10.1145/3025453.3025739
[5] Rebecca Fiebrink. 2015. Wekinator. Retrieved from http://www.wekinator.org/
[6] Rebecca Fiebrink. 2019. Machine Learning Education for Artists, Musicians, and Other Creative Practitioners. ACM Transactions on Computing Education 19, 4: 1–32. https://doi.org/10.1145/3294008
[7] Rebecca Fiebrink. Machine Learning for Musicians and Artists. Retrieved February 23, 2024 from https://www.kadenze.com/courses/machine-learning-formusicians-and-artists/info
[8] Rahel Flechtner and Aeneas Stankowski. 2023. AI Is Not a Wildcard: Challenge for Integrating AI into the Design Curriculum. In Proceedings of the 5th Annual Symposium on HCI Education, 72–77. https://doi.org/10.1145/3587399.3587410
[9] Fabien Girardin and Neal Lathia. When User Experience Designers Partner with Data Scientists. In 2017 AAAI Spring Symposium Series.
[10] Rube Goldberg and Charles Keller. 1979. The best of Rube Goldberg. Prentice-Hall, Englewood Cliffs, N.J.
[11] Lars Erik Holmquist. 2017. Intelligence on tap: artificial intelligence as a new design material. Interactions 24, 4: 28–33. https://doi.org/10.1145/3085571
[12] Rakhee Kallimani, Krishna Pai, Prasoon Raghuwanshi, Sridhar Iyer, and Onel L. A. López. 2023. TinyML: Tools, Applications, Challenges, and Future Research Directions. Multimedia Tools and Applications. https://doi.org/10.1007/s11042-023-16740-9
[13] Janne Van Kollenburg and Sander Bogers. 2019. Data-enabled design: A situated design approach that uses data as creative material when designing for intelligent ecosystems. Technische Universiteit Eindhoven.
[14] Carine Lallemand and Caroline Hummels. 2022. Data-Enabled Design: Hands-on Teaching Activities to Onboard Design Students in the Use of Sensor Data as a Creative Material.
[15] Fatih Kursat Ozenc, Miso Kim, John Zimmerman, Stephen Oney, and Brad Myers. 2010. How to support designers in getting hold of the immaterial material of software. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, 2513–2522. https://doi.org/10.1145/1753326.1753707
[16] Donald A. Schön. 1984. The reflective practitioner: How professionals think in action. Basic books.
[17] Philip Van Allen. 2018. Prototyping ways of prototyping AI. Interactions 25, 6: 46–51. https://doi.org/10.1145/3274566
[18] Maynard Frank Wolfe and Rube Goldberg. 2000. Rube Goldberg: inventions. Simon & Schuster, New York London Toronto Sydney Singapore.
[19] Qian Yang, Nikola Banovic, and John Zimmerman. 2018. Mapping Machine Learning Advances from HCI Research to Reveal Starting Places for Design Innovation. In Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems, 1–11. https://doi.org/10.1145/3173574.3173704
[20] Qian Yang, Aaron Steinfeld, Carolyn Rosé, and John Zimmerman. 2020. Reexamining Whether, Why, and How Human-AI Interaction Is Uniquely Difficult to Design. In Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems, 1–13. https://doi.org/10.1145/3313831.3376301
[21] Elements of AI. Retrieved from https://www.elementsofai.com
[22] KI Campus. Retrieved from https://ki-campus.org/
[23] Udemy. Retrieved from https://www.udemy.com
[24] Teachable Machine. Retrieved from https://teachablemachine.withgoogle.com/
[25] Tiny Motion Trainer by Google Creative Lab. Retrieved from https://experiments.withgoogle.com/tiny-motion-trainer/view/
[26] Edge Impulse. Retrieved from https://www.edgeimpulse.com/
[27] TensorFlow Lite. Retrieved from https://www.tensorflow.org/lite
[28] M5Stack. Retrieved from https://m5stack.com/
[29] Arduino IDE. Retrieved from https://www.arduino.cc/