Building, Analyzing and Interpreting Classroom Engagement: Apps and Machine-Learning Models for an Affordable Programming Education

by

Alpay Sabuncuoğlu

A Dissertation Submitted to the
Graduate School of Sciences and Engineering
in Partial Fulfillment of the Requirements for
the Degree of

Doctor of Philosophy

in

Computer Science and Engineering

KOÇ ÜNİVERSİTESİ

June 19, 2023

Building, Analyzing and Interpreting Classroom Engagement: Apps and Machine-Learning Models for an Affordable Programming Education

Koç University

Graduate School of Sciences and Engineering

This is to certify that I have examined this copy of a doctoral dissertation by

Alpay Sabuncuoğlu

and have found that it is complete and satisfactory in all respects,
and that any and all revisions required by the final
examining committee have been made.

Committee Members:

Prof. Tevfik Metin Sezgin (Advisor)

Prof. Yucel Yemez

Assoc. Prof. Yasemin Kisbu

Prof. Aytac Gogus

Prof. Yusuf Sahillioglu

Date:

ABSTRACT

Building, Analyzing and Interpreting Classroom Engagement: Apps and Machine-Learning Models for an Affordable Programming Education
Alpay Sabuncuoğlu
Doctor of Philosophy in Computer Science and Engineering
June 19, 2023

Ensuring students remain engaged in the classroom is crucial for their success in any given topic. In programming education, promoting hands-on interactions and demonstrating real-world use cases are effective methods to foster engagement. However, schools located in socio-economically disadvantaged areas often lack adequate digital infrastructure, such as computer laboratories, to support such engagement-building tasks. Nevertheless, utilizing mobile devices can support programming education due to their availability and affordability. Additionally, mobile devices can help augment the tangible materials to use in collaborative programming experiences. In this respect, the first goal of my research is to develop an affordable tangible programming education environment using mobile devices that supports the collaborative work of students. On top of building tools to foster engagement, analyzing and interpreting student engagement is also a critical component of the teaching process. Analyzing classroom engagement requires a multi-component evaluation of affective, behavioral, and cognitive states. Yet, limited research has been conducted to create a multimodal classroom engagement dataset and analysis model. To fill this gap, my second goal is to build a multimodal engagement evaluation tool using a single camera to ease teachers' workload in group activities.

Overall my thesis contributes to Human-Computer Interaction (HCI), Artificial Intelligence (AI), and educational technology research areas with six research outputs:

1. An open-source, user-centered, AI-powered, affordable tangible programming environment that was informed by iterative user studies.

2. A set of design considerations for developing paper-based intelligent tangible

programming environments which are informed by iterative and classroom-wide user experience studies.

3. A set of curricular activities to help teachers adapt our programming environment into Turkish national curricula.

4. An open-source audio-visual dataset comprising eight-hour-long video recordings of thirty-three students to predict classroom engagement levels using their self-evaluation scores.

5. Multimodal machine-learning models that can address the multi-component definition of engagement. The image models achieved up to 84% test accuracy on person-based engagement level prediction. The real-time video model that can run on streaming videos achieved 71% test accuracy.

6. A student-centric interactive dashboard to help students view their engagement over time and interpret the results of engagement model prediction.

ÖZETÇE

Sınıfiçi Angajmanın Oluşturulması, Analizi ve Yorumlaması: Ulaşılabilir bir Programlama Eğitimi için Uygulamalar ve Makine Öğrenmesi Modelleri

Alpay Sabuncuoğlu
Bilgisayar Bilimleri ve Mühendisliği, Doktora
19 Haziran 2023

Öğrencilerin sınıf-içi angajmanlarını (ilgi ve motivasyon) sürdürmeleri eğitimlerinin başarısı için büyük önem taşımaktadır. Programlama eğitimindeki güncel örnekler, fiziksel etkileşim içeren aktiviteleri teşvik etmenin ve programlamanın gerçek hayattaki iyi örneklerini göstermenin, öğrencilerin angajmanını artırmak için etkili yöntemler olduğunu göstermiştir. Ancak, sosyo-ekonomik olarak dezavantajlı bölgelerde bulunan okullar genellikle bu tür etkileşimi destekleyecek yeterli dijital altyapıya sahip değildir. Buna karşın, akıllı telefon gibi mobil cihazların uygun maliyetleri ile yaygınlaşması, programlama eğitiminin erişilebilirliğini artırma potansiyeline sahiptir. Ek olarak, mobil cihazlar üzerinde çalışabilecek yapay zeka modelleri, işbirlikli programlama aktivitelerinde kullanılabilecek fiziksel objelerin tanınmasını ve kullanılmasını sağlayabilir. Bu bağlamda, araştırmamın ilk hedefi, öğrencilerin işbirliği içinde çalışmasını destekleyen mobil cihazlar kullanılarak uygun maliyetli bir fiziksel programlama eğitim ortamı geliştirmektir. Sınıf içi angajmanı destekleyecek bu araçları oluşturmanın yanı sıra, sınıf içindeki angajmanı analiz etmek ve yorumlamak da öğretim sürecinin önemli bir bileşenidir. Sınıf içi angajmanın analizi, duygusal, davranışsal ve bilişsel durumların çok bilesenli bir değerlendirmesini gerektirir. Ancak, çok kipli bir sınıf içi angajman veri seti ve analiz modeli oluşturmak için sınırlı sayıda araştırma yapılmıştır. Bu boşluğu doldurmak için ikinci hedefim, grup etkinliklerinde öğretmenlerin iş yükünü hafifletmek için tek bir kamera kullanarak çok kipli bir angajman değerlendirme aracı oluşturmaktır.

Bu tez, İnsan-Bilgisayar Etkileşimi (HCI), Yapay Zeka (AI) ve eğitim teknolojisi araştırma alanlarına altı açıdan katkıda bulunmaktadır:

1. İteratif kullanıcı çalışmalarından elde edilen bilgilerle şekillenen, kullanıcı odaklı,

uygun maliyetli bir akıllı fiziksel programlama ortamı geliştirilmiş ve tüm kaynakları açık-kaynak olarak sunulmuştur.

2. Geniş katılımlı kullanıcı deneyimi çalışmaları gerçekleştirilmiş ve geliştiriciler için akıllı fiziksel programlama ortamları geliştirmek için bir dizi tasarım önerisi sunulmuştur.

3. Öğretmenlerin programlama ortamımızı MEB Programlama müfredatı kapsamında kullanabilmeleri için bir dizi aktivite geliştirilmiştir.

4. Öğrencilerin kendi değerlendirme puanları kullanılarak sınıf içi angajman düzeylerini tahmin etmek için yaklaşık sekiz saatlik sesli ve görsel verilerden oluşan yeni bir çok kipli veri seti toplanmış ve açık kaynak olarak sunulmuştur.

5. Angajmanı çok bilesenli olarak ele alan çok kipli makine öğrenme modelleri geliştirilmiştir. Görüntü modelleri, kişiye dayalı ilgi düzeyi tahmininde %84'e kadar test doğruluk oranı elde etmiştir. Akış halindeki videolarda çalışabilen gerçek zamanlı video modeli ise %71 test doğruluk oranına ulaşmıştır.

6. Öğrencilerin angajmanlarını zaman içinde görüntülemelerine ve model çıktılarını yorumlamalarını destekleyecek, öğrenci odaklı etkileşimli bir kontrol paneli geliştirilmiştir.

ACKNOWLEDGMENTS

PhD is a long journey, and I am grateful I met with the most supportive individuals and organizations through this journey. First and foremost, I would like to express my sincere gratitude to my advisor, Prof. Tevfik Metin Sezgin. Your guidance, expertise, and unwavering support have been instrumental in shaping the direction of my research and in my personal and professional growth. I have been fortunate to have you as my advisor, and I am grateful for your mentorship.

I am deeply grateful to my wife, Dr. Ceylan Besevli Sabuncuoğlu, for her constant encouragement, understanding, and love throughout this challenging journey. Your support is a constant source of motivation. My mother, Emine Kuşgöz, father, Yavuz Sabuncuoğlu, and sister, Nilay Sabuncuoğlu have always been there for me, and I am thankful for their love and support.

I extend my heartfelt appreciation to the members of my thesis jury: Assoc. Prof. Yasemin Kisbu Sakarya, Prof. Yücel Yemez, Prof. Aytaç Göğüş, and Prof. Yusuf Sahillioğlu. Your insightful feedback, constructive criticism, and valuable suggestions have significantly enriched my research work.

I would also like to acknowledge the entire Intelligent User Interfaces Lab and its members for their collaboration, camaraderie, and stimulating discussions. The collective efforts of the lab group have contributed to a vibrant research environment, fostering creativity and innovation. Additionally, I would like to extend my gratitude to the KUAR Lab members, with special mention to Yağmur Kocaman, Berk Göksenin Tan, and Cansu Çetin. Your contributions to my data collection studies were invaluable, and I appreciate your dedication and assistance.

I would like to express my thanks to TUBITAK (project no 218K436) and KUIS AI Center for their generous support and funding, which enabled me to carry out my research work effectively. I would also like to thank Meryem Şeyda Özcan for her contributions to the data collection and analysis process of our TUBITAK project. We visited schools together and structured the curriculum studies together. I am also grateful to the teachers and students who participated in my studies and provided valuable feedback. I would also like to all NGOs and organizations that supported my research work, including the Twin Science and Robotics, Rahmi M. Koç Müzesi (Yeşim Öztüran), and Türk Eğitim Gönüllüleri Vakfı.

Lastly, I would like to acknowledge all the individuals, colleagues, friends who have supported me in various ways throughout this challenging endeavor. Your encouragement, understanding, and belief in me have been instrumental in my success. I am sincerely grateful to each and every person and organization mentioned above for their contributions, guidance, and support, without which the completion of my PhD dissertation would not have been possible.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABBREVIATIONS

BT	Boosted Trees
CEI	Classroom Engagement Inventory
CNN	Convolutional Neural Network
CT	Computational Thinking
DL	Deep Learning
EBM	Explainable Boosted Machine
GDPR	General Data Protection Regulation
ICT	Information and Communication Technology
IUI	Intelligent User Interfaces
kNN	k-Nearest Neighbors
KVKK	Kisisel Verileri Koruma Kurulu
LIME	Local Interpretable Model-Agnostic Explanations
LSTM	Long-Short Term Memory
LR	Logistic Regression
ML	Machine Learning
NGO	Non-Governmental Organization
PCA	Principal Component Analysis
RBF-SVM	Radial Basis Function Support Vector Machine
RF	Random Forest
TUI	Tangible User Interfaces
UNESCO	United Nations Educational, Scientific and Cultural Organization

LIST OF PUBLICATIONS

Venue	Title
ACM TEI'20	Tangible music programming blocks for visually impaired children [Sabuncuoglu, 2020]
ACM IUI'20	Kart-ON: affordable early programming education with shared smartphones and easy-to-find materials [Sabuncuoğlu and Sezgin, 2020]
IEEE Vis 2021 - Vis-Activities Workshop	Teaching K-12 Classrooms Data Programming: A Three-Week Workshop with Online and Unplugged Activities [Sabuncuoğlu et al., 2021]
FabLearn 2022	Prototyping Products using Web-based AI Tools: Designing a Tangible Programming Environment with Children [Sabuncuoglu and Sezgin, 2022c]
ACM EICS 2022	Kart-ON: An Extensible Paper Programming Strategy for Affordable Early Programming Education [Sabuncuoglu and Sezgin, 2022b]
ACM CHI 2023 Workshop on Child-centred AI Design	Towards Building Child-Centered Machine Learning Pipelines: Use Cases from K-12 and Higher-Education [Sabuncuoglu et al., 2023a]
Preprint	Exploring Children's Use of Self-Made Tangibles in Programming [Sabuncuoglu and Sezgin, 2022a]
In-submission	Developing a Programming Environment for Affordable Mobile-Powered Education: App and Curriculum
Technical Report - Preprint	Multimodal Group Activity Dataset for Classroom Engagement Level Prediction [Sabuncuoglu et al., 2023b]

EICS 2023	Developing a Multimodal Classroom Engagement Analysis Dashboard for Higher-Education Students [Sabuncuoglu and Sezgin, 2023]
In-Submission	Multimodal Group Activity Dataset for Classroom Engagement Level Prediction

Chapter 1

INTRODUCTION

In today's rapidly evolving world, honing 21st-century skills such as *computational and creative thinking* is a must. Programming education in the early years lays a crucial foundation for the development of these skills [Grover and Pea, 2013]. In a traditional programming education curriculum, students learn building algorithmic structures via hands-on coding activities and develop their computational thinking skills. (*Computational thinking* (CT) includes problem-solving techniques used in computer science, such as abstraction, decomposition, algorithmic thinking, and pattern recognition [Wing, 2006]. Applying and refining these techniques in real life early on has a significant impact on future careers and the lifelong success of students [UNESCO, 2019].

To empower students with computational thinking (CT) skills, it's not enough to merely teach programming; we must keep them *cognitively, behaviorally*, and *affectively engaged*. This three-component engagement, referred to as *classroom engagement*, means students' active involvement in learning, discussion, and reflection with *peers, teachers,* and *materials* in a classroom environment [Fredricks et al., 2004]. It is a key factor in predicting students' success and future motivation in a given course [Fredricks et al., 2004]. Therefore, in designing solutions that support students' CT skills, building engagement with (I) the right set of tools and (ii) analyzing engagement by combining multimodal data is crucial. Additionally, affordability is another important consideration in designing such solutions, as most economically disadvantaged communities lack digital infrastructure, such as computer laboratories that are commonly employed in programming education and CT skills development. Given these factors, *delivering an affordable programming plat-*

form that can engage students in programming activities and support teachers in tracking classroom engagement throughout these activities has become a trending research and development interest.

Within this context, this chapter outlines the research gaps and my motivation (Section 1.1), followed by the research questions (Section 1.2) and contributions (Section 1.3). Then, I provide an overview of the methodology pursued throughout the thesis (Section 1.4), give an overview of ethical considerations and data collection regulations (Section 1.5) and present a snapshot of the dissertation flow (Section 1.6).

1.1 Research Gaps

Gap 1: *Existing programming environments are not accessible to the economically disadvantaged communities due to the lack of sufficient resources for establishing and maintaining adequate computer laboratories [Kraemer et al., 2009, OECD, 2020]. While the wide availability of mobile devices [Jacobs, 2013] can increase the affordability of programming tools by increasing collaborative work and augmenting the capabilities of tangible materials, limited research focused on using mobile devices in a collaborative classroom environment. In addition, the existing solutions offer limited input/output variations in the semantic design and do not demonstrate a similar syntactic structure with current popular scripting languages, which can cause frustration while transitioning into real-life coding [Moors et al., 2018, Parsons and Haden, 2007].*

With the recent advances in processors, internet connection, and cameras, smart mobile devices (phones and tablets) achieved the capacity to augment the capabilities of physical materials. Using physical materials, such as paper programming cards, can enhance collaboration and engagement by supporting the active sharing of the learning materials in group activities [Marshall, 2007, Zuckerman et al., 2005]. I developed an affordable, paper-based mobile-powered programming environment, *Kart-ON*, that presents a high-level tangible command set that can be recognized using computer vision algorithms. I regularly met with teachers throughout Kart-ON's design and evaluation process. Kart-ON utilizes *p5.js*, a popular web creative coding framework, and maintains a similar syntactic design to help students and teachers transition from paper to digital programming. I evaluated the impact of our platform and curriculum with small and large-scale evaluations with UX and curriculum studies.

Gap 2: *Existing work on engagement recognition covers only a limited presentation of an active classroom environment, which restricts the development of accurate engagement prediction models. The need for a comprehensive, multimodal, and publicly-available dataset is acknowledged, yet the effort given in this direction is inadequate [Dewan et al., 2019]. Most of the video datasets have been collected in an online setting, where students interact with an interface while a camera faces directly toward the participants [Delgado et al., 2021, Yu et al., 2021]. Due to this limitation, the existing models for engagement recognition misses the multi-component (emotional, behavioral and cognitive) nature of classroom engagement.*

I collected group activity video data with a focus on representing the multi-component definition of engagement in multiple learning styles. The dataset is a valuable resource as it captures engagement in a multimodal representation with the frame-by-frame self-evaluation of students. I trained a set of face and video-based deep learning models using recordings totaling seven-and-a-half-hour-long. The dataset is an initial step towards developing a multimodal model that can effectively predict classroom engagement using a single camera while considering the multi-component definition of engagement.

Gap 3: *Many research-based strategies, curricula, and tools cannot find a place in classroom practices due to the lack of student and teacher-centric approaches [Antle and Hourcade, 2022]. Educational technology research needs open-source and easy-to-customize research outputs to help teachers easily maintain classroom flow and build student engagement.*

In each stage, I developed student and teacher-centric activities, tools and dashboards to reduce the teacher workload and enhance student motivation and collaboration. Throughout my PhD, I met with teachers and students to develop in-classroom tools and activities to ease the moderation of programming education. I developed a *twelve-week curriculum* based on teachers' feedback to help them adopt Kart-ON as the main programming environment. Additionally, I developed an *interactive dashboard* to support teachers and students in analyzing and interpreting the engagement data in a student-centric way. By providing teachers with the necessary tools and support, my thesis aims to bridge the gap between research outputs and classroom practice.

With these motives, I focused on building an engaging, affordable, and collaborative K-12 programming education environment and developing intelligent systems to analyze and interpret students' classroom engagement. I utilized recent deep learning (DL) research techniques while developing the mobile-device-powered solutions and multimodal engagement analysis model. My contributions to date have been benefiting the stakeholders of the K-12 education ecosystem, including students, teachers, policy-makers, and NGOs, as well as HCI, AI, and Educational Technology fields.

1.2 Research Questions

Throughout my PhD studies, I aimed to answer the following three main research questions:

- **(RQ1)** How can we design affordable programming environments using the advantage of mobile devices to enhance collaboration and engagement in crowded classrooms?

- **(RQ2)** How can we analyze engagement automatically in a multimodal fashion using already-available mobile device cameras in the classroom?

- **(RQ3)** How can we inform the design of classroom tools to help teachers maintain the classroom flow and support students in interpreting their engagement?

These three questions led us to develop several tangible research outcomes that we published open-source. In the next section, I summarized my main contributions and outputs.

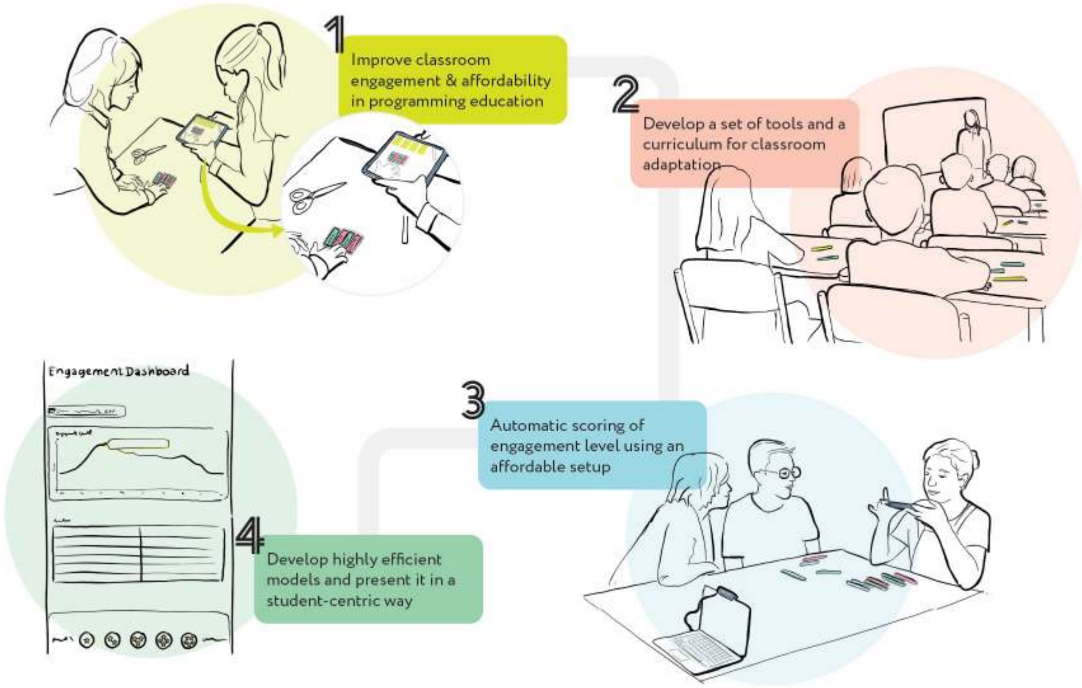


Figure 1.1: **Overall flow of the research motives:** From building an affordable and engaging programming environment to analyzing and interpreting students' engagement in a student-centric way, my main motive was to develop intelligent systems that can support teachers in delivering equal education for all.

1.3 Main Research Contributions

My thesis contributes to Human-Computer Interaction (HCI), Artificial Intelligence (AI), and educational technology research areas in six ways:

- I developed an open-source, user-centered, AI-powered, affordable tangible programming environment (Kart-ON) informed by iterative user studies with ICT teachers and middle school students.
- I conducted user experience (UX) studies with middle school students, which led me to develop a set of design considerations for developing paper-based intelligent tangible programming environments.

- I developed a twelve-week programming education curriculum which was evaluated by a large-scale curriculum study.
- I collected and open-sourced a new audio-visual dataset to predict classroom engagement levels using their self-evaluation scores.
- I built multimodal machine-learning models that can address the multi-component definition of engagement. The image models achieved up to 85% test accuracy on person-based engagement level prediction. The real-time video model that can run on streaming videos achieved 71% test accuracy.
- I developed an interactive dashboard by conducting focus group studies to help students view their engagement over time and discover their learning patterns.

1.3.1 Kart-ON: An Affordable and Engaging Programming Platform

One of our main contributions is the Kart-ON programming environment, an affordable system providing a set of predefined paper programming cards allowing users to complete different creative coding tasks. Kart-ON can be defined as an environment rather than an application, as it provides a set of web and mobile tools with tangible objects customized for teachers and students. The smartphone application recognizes physical cardboard blocks by utilizing the recent deep mobile vision techniques (i.e., text recognition and on-device transfer learning). Combining these recognized cards, students can create new functions that can be represented with any custom tangible object. Representing an abstract programming element using custom everyday objects, i.e., representing a touch-based circle drawing function with a tennis ball as seen in Figure 1.2-f, reduced the total screen time of smartphones. In our user studies, an average group activity took only around five minutes in a forty-five-minute lecture time. The reduced screen time allowed groups of students to share their smartphones in classrooms effectively. Figure 1.2 summarizes the overall workflow of the Kart-ON mobile application and interaction with tangible objects.

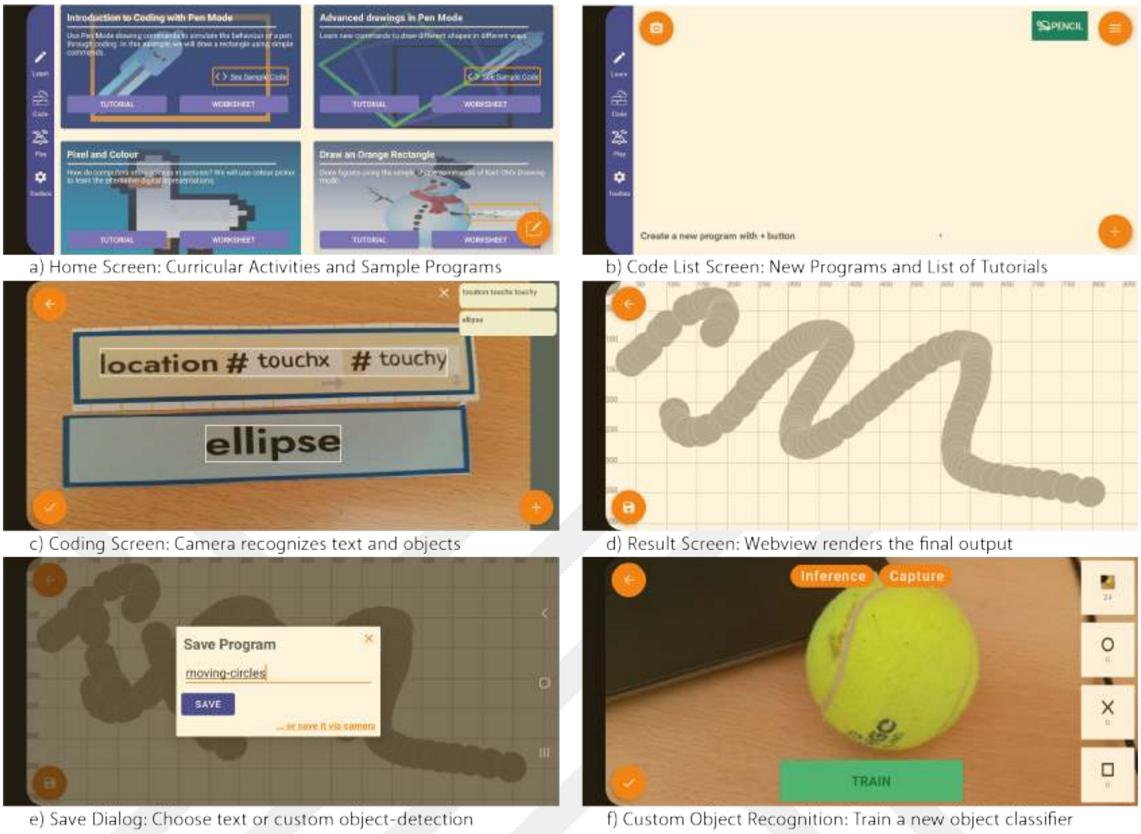


Figure 1.2: **Kart-ON mobile application main screens:** (a) Home screen contains curricular activities. (b) Code List screen shows the saved and sample programs. (c) Coding screen opens the camera to recognize command cards. (d) Result screen renders the code output. (e) Save dialog presents text and custom object options. (f) Custom object recognition takes photos to detect the object in later uses.

1.3.2 The Classroom Engagement Dataset and DL Models

The multimodal classroom engagement data includes seven and a half hours of recordings from seven different higher-education student groups (34 individual participants.). The 'classroom' in the dataset name resonates my intention to build a dataset that can be used for real classroom environments. But, the dataset contains videos recorded in controlled environment and mimicked the activities taking place in classrooms. The dataset in total contains 26540 frames and self-evaluations of each individual participant for each frame. Starting the process with higher-education students provided two significant advantages: (i) It accelerated our collection progress, as getting approval from the ethical committee and university is faster than getting the necessary permissions from middle schools, and (ii) Publishing the data of adults participants in open-source format requires less regulations compared to publishing children's data. Collecting data from children requires careful consideration at every step [UNESCO, 2021]. Therefore, we decided to collect the data from adult students and fine-tune our final model with data from middle school students.

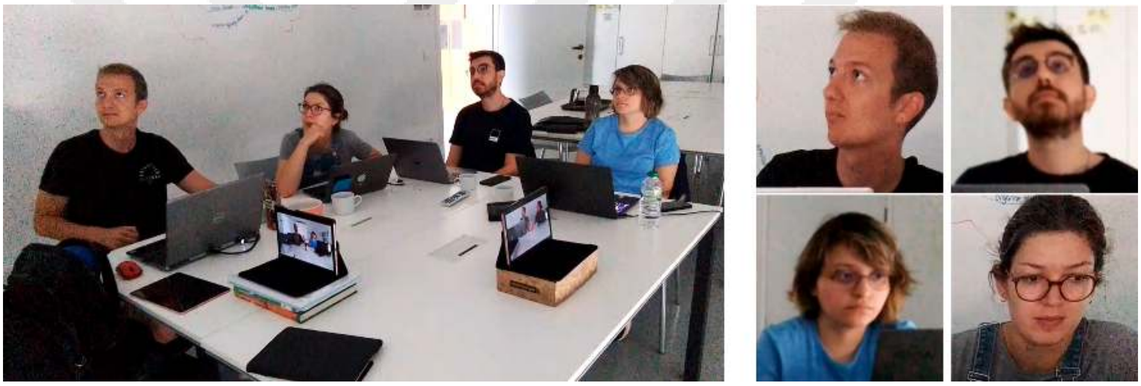


Figure 1.3: **A glimpse of the Classroom Engagement Dataset:** The left image shows a sample frame from group view recordings. We run the OpenPose on these frames. The tile of four face images show individual faces extracted using MediaPipe and FFmpeg frameworks.

We collected the video data of the students as they learned a creative coding library and conducted digital and physical activities using this library. Groups of students worked together in the tutorial-following and activity-completing steps. Figure 1.3 shows a sample frame from our audiovisual recordings. Then, students self-evaluated themselves and annotated their engagement for each frame. After collecting our audiovisual data and engagement-level annotation, I extracted facial, positional features and analyzed them with my open-sourced pipeline. To our knowledge, this end-to-end, video-to-interpretable feature extraction pipeline is the most comprehensive multimodal open-source learning analytics data pipeline that utilizes state-of-the-art deep learning techniques. In these experiments, I observed that even a small amount of data with combined face, pose, and text features can effectively predict engagement levels. Yet, I have not developed a generalizable model that can classify the engagement levels of different participants' engagement levels.

After collecting, annotating and analyzing the data, I started developing deep learning models to predict categorical engagement levels using two different inputs. The first input is the individual face-area, and the second input is the group activity video slices. These models are built upon the recent state-of-the-art deep learning models for object and video action classification. The face-based models achieved accuracy between 0.55 and 0.84 for for personal engagement level classification. The video-based models achieved up to 0.74 accuracy for categorical group-level engagement classification.

1.3.3 Classroom Activities and Tools

The practical impact of developing a programming environment, collecting a dataset, and building deep learning models can only be truly realized if teachers are willing to use them. To ensure that my research outputs get utilized in classrooms, I have developed user-friendly helper tools that are easy to learn, use, and customize. In the course of this development, I conducted interviews and focus group meetings with both teachers and students.

While Chapter 5 and 8 mainly focus on increasing the student-centric adaptation of the research outputs, this document also provides various examples of my efforts in developing helper tools. For instance, Kart-ON Pen Mode was created to help teachers introduce programming commands to students in a step-by-step manner that matches the behavior of a pen agent with drawing commands. But, students and teachers showed a strong interest in the tool, which led us to integrate this tool into the first three weeks of the programming curriculum.

Kart-ON Helper Tools and Set of Activities: To ease the adoptation of the programming environment and accelerate the semester-long use, I developed a twelve-week illustrative curriculum that can be easily adapted for paper programming environments. Our analysis of UX studies and ideation workshop guided the educational content development process. I developed our illustrative curriculum in line with the current Turkish programming education curriculum for K5-6 grades. The first two weeks of the curriculum consist of unplugged activities that focus on basic problem-solving methods. The following three weeks use Pen Mode, an extension of the Kart-ON programming environment using a similar syntax to Papert's LOGO, to teach sequential algorithm building and loop structure. After students grasp the fundamentals of programming and gain experience with computer vision tools, the curriculum introduces the basics of computer vision and object recognition in a game-like environment for two weeks. In the rest of the curriculum, they use Drawing mode, where they apply their creative coding skills with more complex programming structures like variables, functions, and branching.

Partnering with *Koc University Independent Evaluation Lab*, we tested our curriculum with three hundred students in three different schools. We measured students' development in computational thinking skills using pre and post-tests. We analyzed the results to propose an integration plan for our programming environment and curriculum into Turkish programming education. I did not include the curriculum test process and results to maintain the focus of this thesis.

Engagement Dashboard: I designed the dashboard to motivate students to inquire about their learning habits and support teachers in thoroughly gauging classroom engagement. Making classroom engagement visible can help students to understand why and where they failed to grasp the concept and how they can succeed following their more engaging moments. Teachers can analyze their classrooms' engagement through different activities and manage their classroom flow accordingly. Considering these possible outcomes, I designed a dashboard that contains four main components: (1) Information Area, (2) Engagement Score Chart, (3) Score Prediction Model's Rules, and (4) Cognitive Engagement Quiz Area. The *information area* describes the learning task. *Engagement score* chart visualizes the scores predicted by our baseline multimodal classifier for classroom engagement. It informs participants about the *score prediction model's rules* to make the DL model more interpretable. To develop a dashboard that can help students self-evaluate themselves by considering their cognitive engagement, *cognitive engagement quiz area* generates a quiz that automatically assesses students' confidence based on the information area's description. After the initial wireframe prototype of the dashboard, I dynamically developed the dashboard on Observable[Observable, 2023] through two user studies.

1.4 Research Methodology: Student-centric, Iterative Design

Throughout my Ph.D. studies, I consistently engaged with students and teachers to design tools, activities, and systems that prioritize student-centeredness. Overall, I engaged with students (Ntotal=459) and ICT teachers (Ntotal=53) at multiple time points of the research.

The development of Kart-ON involved a thorough iterative process that entailed conducting UX studies with over one hundred and fifty students in individual, group, and classroom settings. To ensure ethical requirements were met, the classroom engagement data collection, analysis, and interpretation were also progressed iteratively. A summary of all the user studies conducted during my PhD studies is provided in Figure 1.4.

The development of Kart-ON was iterative. I periodically met with both K5-6 grade students and teachers. During and after the pandemic, I had the chance to observe both students' and teachers' needs and experiences in different settings. I conducted our studies through the pandemic in open areas like parks and other common meeting areas. When the schools started allowing researchers to conduct classroom studies, we completed our in-classroom UX surveys and long-term curriculum studies. We reported our quantitative and qualitative analysis of this series of workshops (approximately with more than one hundred and fifty students) and design considerations for future paper-programming environment developers in our [Sabuncuoglu and Sezgin, 2022b] paper. These considerations can help researchers to design and evaluate the syntax, abstraction level of programming elements, card

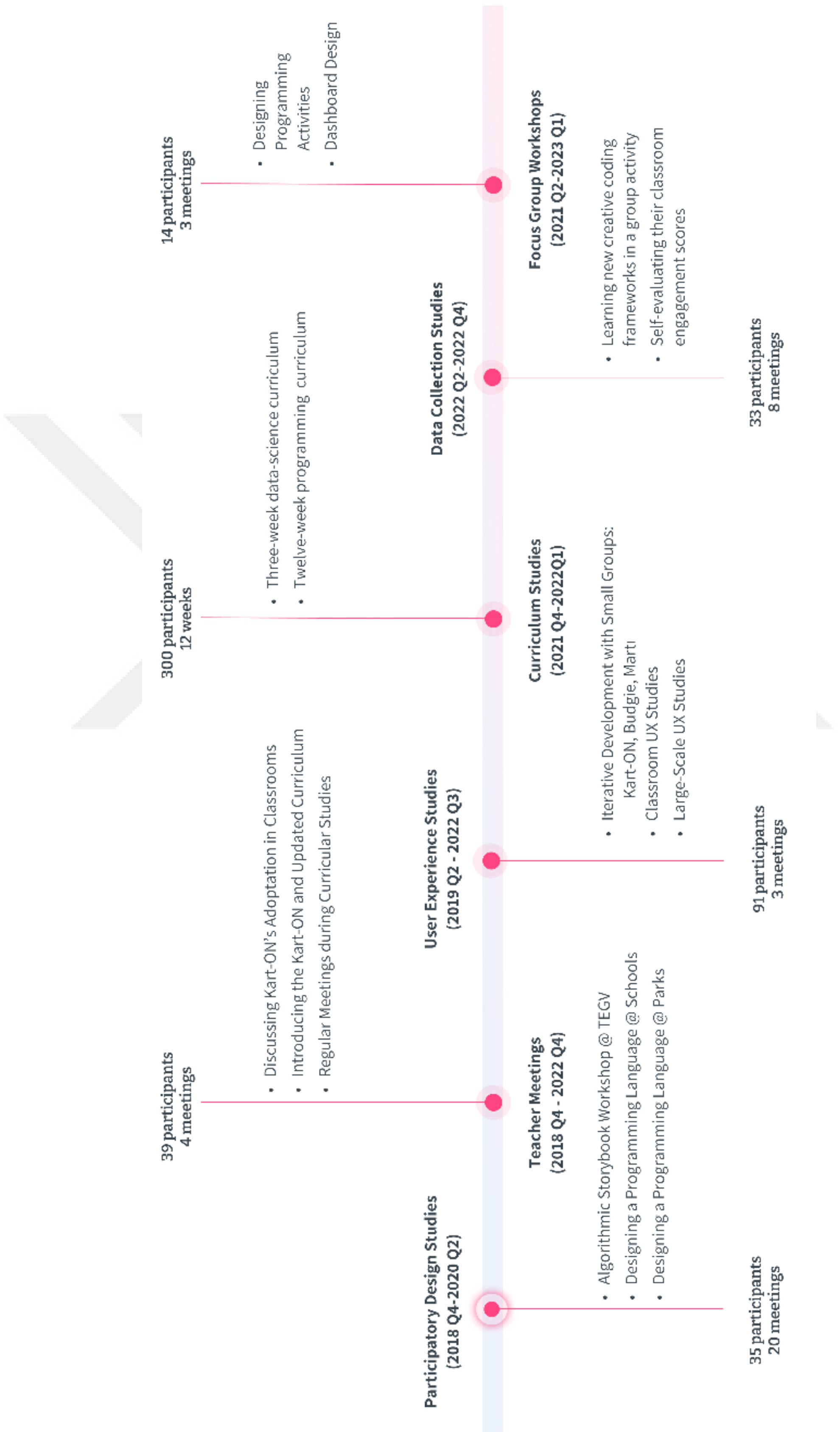


Figure 1.4: **Summary of all user studies**: A simplified timeline of user studies conducted throughout my Ph.D. studies using quarterly blocks. I conducted a total of fifty user studies with more than five hundred participants.

interaction methods, mobile-tangible interaction durations, and individual-group use differences of future paper programming environments. I shared all the code of Kart-ON and its extensions, open-source, at Kart-ON's Github Repository

In the *Classroom Engagement Data* collection process, I consistently upheld my commitment to considering ethical requirements at every stage. While the iterative process was similar to my previous development processes, I ensured the learning activity design did not deviate significantly after the first two data collection sessions to ensure that all data could be utilized in the model training stage. The activities revolved around a common theme, where all groups engaged in creative coding activities while watching YouTube tutorials, followed by hands-on activities related to online learning content. In the data collection stage, I aimed for students to improve their knowledge while becoming participants in a scientific study. After the data collection process, I conducted a focus group with the students to design an interactive dashboard collaboratively. This enabled gathering insights into their preferences and needs, which I incorporated into the final design. Ethical considerations remained a priority throughout the process, and we ensured that the students' data privacy was protected. The resulting dashboard was user-friendly and easy to navigate, providing an effective tool for visualizing and analyzing the collected data.

1.5 Ethics Statement

Throughout my Ph.D. studies, I studied and worked closely with more than four hundred and fifty students, fifty teachers, and many other stakeholders of education. Involving human participants in the prototyping process of an experimental product requires careful attention to participants' health, well-being, and privacy. For each study involving human participants, I obtained ethical approval for this study from **Koç University Committee of Human Research**. In this chapter, I briefly explained the overall ethical considerations throughout my Ph.D. studies and described the participation process for each chapter.

1.5.1 Involving Children into the Design Process

As I elaborated more detail in Section 2.3, children can adopt different roles in the design process, which Druin categorized into four: user, tester, informant, and design partner. Throughout the user studies, children have been involved in the design processes taking all these different roles.

Small-Scale Studies

I contacted the parents via the researcher's personal contacts and snowballing. The parents have informed the study details via Whatsapp and reported their consent via this channel. The studies under COVID-19 regulations could not be conducted collaboratively. I met with children in the parks and other public open spaces individually. Before and after COVID-19 regulations, I met with them in groups in their schools. All the small-scale studies are conducted in Antalya and Istanbul, major cities in Turkey. Children participants had a mixed understanding of what programming and AI can or cannot do. Most of them use smartphones/tablets daily and exhibit digital literacy skills.

Large-Scale Studies

For the large-scale UX and curriculum studies, I contacted the District Education Office of Sarıyer and the City Education Office of Istanbul to obtain permission to conduct the studies in the schools (Appendix D). Following their guidance, I contacted the teachers by visiting their schools and informed them about the study details. Before conducting each study, we worked together and updated the materials if needed. In these studies, teachers informed parents about the study details. Similar to the small-scale studies, we obtained parents' and students' consent with paper forms (Appendix E).

1.5.2 Data Protection and Privacy

All study data is protected by KVKK[1] and GDPR[2].

Children's Data

All children's data is collected anonymously. Even the researchers have no access to children's identities, we informed parents about the data protection and privacy rules. Their data is protected in cloud storage with two-factor authentication. No children's images or videos were collected throughout the development of the evaluation process.

Adults' Data

I conducted studies together with teachers, policy-makers, NGO staff, and university students. I did not collect any personally identifiable data (PID) when they participated in our studies as testers or commentators. I took notes or asked them to fill out some surveys.

The only PID is collected during the collection of the *Classroom Engagement Dataset*, where I met with university students. The data collection, storage, and sharing process comply with the GDPR. We shared the data publicly only if the participants approved the publishment of data. We offered levels of data protection and privacy setting for their approval, which they can choose from:

- Sharing video and sound recordings.
- Sharing only sound recordings,
- Sharing only video,
- Sharing transcripts

[1] `https://www.kvkk.gov.tr/en/`

[2] `https://gdpr-info.eu/`

- Sharing anonymized deep learning model embeddings,

Following their preferences, the publicly shared *Classroom Engagement Dataset* only includes the video and face recordings of the participants who gave consent (Appendix F).

1.6 Dissertation Overview

Chapter 2 summarizes the background of this thesis and reviews the literature on the current scene of programming education, tangible programming interfaces, and analyzing classroom engagement.

Chapter 3 describes the design, analysis and findings of the exploratory design studies that informed my programming environment design process.

Chapter 4 explains the app development process, paper card design, mobile vision recognition architecture and UX study results with our final design considerations.

Chapter 5 describes our twelve-week curriculum based on our focus group meeting with teachers.

Chapter 6 explains the data collection and analysis process, where I followed an iterative approach to actively update the data collection steps by analyzing the previous data.

Chapter 7 presents the baseline deep learning architectures for the engagement-level classification task using images and videos.

Chapter 8 describes the details of the student-centric interactive dashboard which is specifically aimed at students that have no expertise in AI to help interpret their engagement levels.

Chapter 2

BACKGROUND

This section summarizes the existing research on programming education focusing on the use of tangible programming environment, its challenges and opportunities. Then, overviews the design process for and with children. Next, it defines classroom engagment and briefly describes the recent advances in engagement recognition. Finally, it summarizes state-of-the-art computer vision (CV) architectures that can inform the design of video-based engagement recognition models.

2.1 Current Programming Education Scene

Teaching programming and developing digital literacy have been integrated into primary education in most developed countries [Grover et al., 2015]. Previous research demonstrated that participating in programming activities supports children in developing computational and creative thinking skills [Hubwieser et al., 2015]. *Computational thinking* (CT) encompasses a set of problem-solving techniques that have been used in computer science, such as abstraction, decomposition, algorithmic thinking, and pattern recognition. Computational thinking has also been regarded as a conceptual framework incorporating the various skills required for 21st-century subjects such as mathematics, engineering, or science [Wing, 2006].

The first efforts to develop CT skills in early childhood are based on Papert's work on LOGO language, which was the first language that was built for educational purposes and focused on developing CT-related concepts. [Papert, 1999]. Since then, programming has been a subject of discussion for the K-12 curriculum. In recent years, most countries have integrated programming education into their primary or secondary school curriculum. These curricula have been utilizing different kits and tools to introduce basic programming concepts. Although the traditional input/output (I/O) methodology to practice programming is using personal computers, different I/O modalities can be utilized to feed inputs and produce outputs throughout the coding process. From the I/O modality perspective, we can classify the existing programming education tools into three categories: (1) Visual web-based interfaces, (2) Tangible/robotic kits, and (3) Unplugged activities.

Visual web-based interfaces use a drag-and-drop type of interaction to code programs visually. These programs can be stories and animations. Teaching programming via web-based programming environments that support drag-and-drop interaction has become the prominent approach among education communities, and recent curricula examples generally adopt activities using these platforms [Hubwieser et al., 2015]. Scratch is the most commonly used one, developed by the Lifelong Kindergarten group of MIT Media Lab [Maloney et al., 2010]. The Turkish national curricula and most other countries promote the use of similar visual interfaces to teach programming via block-based style [Hubwieser et al., 2015]. However, these platforms are not affordable as it requires one computer per student. Although most have access to computers at home in developed countries, not every child has access to individual computers in many countries [OECD, 2022]. The 2022 OECD report showed that 29% of children younger than 15 in Turkey do not have computers at home. The computer infrastructure of schools shows a similar trend, in which developing countries have less access to computers at school [OECD, 2022]. Even if the schools have access to computers, their computing power might not be enough to run the applications. Therefore, finding ways to teach young children essential future work skills without personal computers is crucial.

The utilization of smartphones and tablets can enhance the accessibility of programming education [UNESCO, 2019]. According to Qualcomm's report on education modernization, the massive number of mobile connections worldwide, surpassing 6.3 billion, presents an exceptional opportunity to revolutionize 21st-century education. These devices have evolved into the largest information and communication platform in history [Jacobs, 2013]. However, incorporating them into educational tasks poses challenges [Hirsh-Pasek et al., 2015]. For instance, Microsoft's Touch

Develop was a programming environment specifically designed for mobile devices, utilizing touch interaction, sensory input, and multimodal output [Tillmann et al., 2012]. Nevertheless, limitations such as small screen size and the absence of a physical keyboard restrict interaction capabilities. Another obstacle lies in integrating mobile education with learning theories, an area that has received less attention compared to other digital platforms. Hirsh-Pasek et al. propose four key principles for classifying an application as educational: *active*, *engaging*, *meaningful*, and *socially interactive* [Hirsh-Pasek et al., 2015]. Within this framework, *active learning* refers to being mentally and physically involved during the learning process, while *engagement* involves maintaining focus on a task. *Meaningful learning* occurs when children grasp the underlying logic of a subject and connect new information to their existing knowledge. Lastly, *social interaction* fosters the sharing and reflection of knowledge with peers. The challenges posed by small screen sizes and the absence of external input/output devices hinder the adoption of these devices in education. However, we believe that by enabling tangible experiences through mobile devices, we can overcome these challenges and effectively incorporate these four principles into educational applications.

Tangible User Interfaces (TUI) enable hands-on interaction and provide physical artifacts for controlling digital features [Ishii and Ullmer, 1997]. These interfaces prioritize active physical engagement in both input and output spaces. Research in education and psychology indicates that incorporating tangible elements in learning environments promotes active, collaborative, and inclusive learning experiences [Zuckerman et al., 2005, Marshall, 2007]. Recognizing these advantages, tangible programming tools have been integrated into programming classrooms [Horn and Jacob, 2007b]. From a technical standpoint, tangible programming tools can be classified into three categories: electronic, unplugged, and digitally augmented paper-programming kits. Commercially available electronic tangible programming tools [Arduino, 2021, Ball et al., 2016, Twin Science and Robotics, 2021, LEGO, 2021, MBlock, 2021] can be programmed either directly on the device or via a computer. Despite evidence supporting the effectiveness of these tools in programming education, their high cost and logistical challenges make them inaccessible to most students and schools. In the following section (Section 2.2), I provide a more comprehensive overview of TUIs since our programming environment also employs tangible programming cards as input.

Another category of tools in programming education is unplugged activities [Bell and Vahrenhold, 2018]. Unplugged activities use everyday materials to complete algorithmic tasks without requiring an electronic device. In these activities, students generally mimic the behavior of an algorithmic agent and execute the algorithm step by step in a creative drama setting. Nevertheless, these tools have limited capacity to demonstrate high-level programming concepts, which can restrict future technology-career interest [UNESCO, 2015].

Between these two categories of programming tools, unplugged activities and computationally enhanced tangible programming environments, paper-programming tools [Fuste and Schmandt, 2019, Deng et al., 2019, Goyal et al., 2016, Horn and Jacob, 2007a, Hu et al., 2015, Sabuncuoğlu et al., 2018, Sabuncuoğlu and Sezgin, 2020, Im and Rogers, 2021, Tada and Tanaka, 2015] stand out as they provide a high-level tangible command-set that can be recognized using computer vision algorithms. These tools utilize mobile or stationary computer vision systems to recognize paper-printed command blocks. They have been tested in a domain-specific manner, primarily resulting in game-like activities or robotic control. For instance, "Hyper Cubes" utilizes high-level programming cubes to control the algorithmic flow of stick figures in an AR development environment [Fuste and Schmandt, 2019]. On the other hand, Tada et al.'s Sheets offer a stationary system with a lower-level command set designed for robotic applications. Nevertheless, this system has limited mobility and abstraction capacity [Tada and Tanaka, 2015].

However, there are two significant limitations of these paper-based tangible programming tools. Firstly, these domain-specific tools have limited support for input and output variations. They do not allow for the use of custom tangible blocks to define new functions. Secondly, the semantic design of these technologically appealing tangible blocks does not demonstrate a similar structure with current popular scripting languages, which can cause frustration during the transition to real-life coding [Moors et al., 2018, Parsons and Haden, 2007].

2.2 Benefits and Challenges of TUIs in Programming Education

Developing an impactful and engaging tangible programming environment requires thorough understanding of the context and processes that can impact a child's cognitive development. Jean Piaget and Seymour Papert exerted a significant impact on the structure and pedagogy of programming education. Piaget's constructivist theory argues that children learn through an active process of experimenting and play, which leads to building upon the knowledge they have already acquired [So, 1964]. Building upon Jean Piaget's constructivism theories, Papert introduced the concept of *constructionism*, which explains that children's learning process best occurs when actively constructing mental models in their minds, utilizing their experiences and actions. Building upon these fundamental theories, the concept of Tangible User Interfaces (TUIs) is rooted in the idea that we are naturally skilled at manipulating objects in three dimensions, and tangible programming uses physical objects to create mental, algorithmic models of a programming activity.

TUIs utilize physical tokens to control digital interfaces, merging the richness of the physical world with human-computer interaction [Ishii and Ullmer, 1997]. When adapting TUIs in programming, the interface allows using physical objects or manipulatives to represent and manipulate data and control structures in programming activities. It is often used as a teaching tool to help students learn programming concepts in a more hands-on and interactive way. In tangible programming, students might use physical blocks or cards with symbols on them to represent different commands or data values in a program. They might then arrange these blocks or cards in a logical order to create a program, which can then be run on a computer or other device. This visual representation help students develop a better understanding of how programs work and how different pieces of code fit together [Horn and Jacob, 2007b]. Research has shown that tangible programming can be beneficial for understanding programming systems, as well as connecting the learner to their physical reality, as opposed to screen-based systems [Marshall, 2007].

In the last two decades, the tangible programming field has been observing a growing interest in developing programming tools for educational use [Horn et al., 2008, Horn and Jacob, 2007c, Raffle et al., 2004]. The classroom studies of tools demonstrated that using TUIs shows great promise in teaching computational thinking concepts [Kafai and Burke, 2014]. TUIs help students develop classroom engagement by fostering hands-on-learning, collaborating and actively discussing with other students, and improving accessibility:

- **Hands-on learning:** Tangible programming tools allow students to learn programming concepts in a more hands-on and interactive way, rather than just reading about them or watching a video. This can make the learning process more engaging and enjoyable for students.

- **Visualization:** The physicality of the tools can help students visualize how different pieces of code fit together, making it easier to understand how programs work. The interface elements can keep children mentally active by engaging multiple senses in the constructive learning process.

- **Collaboration:** When the tools can be used in group settings, they support students to collaborate and work together to create programs. This can help students develop important social and communication skills. The three-dimensional form of the tangibles can support natural group interaction in a shareable workspace and increase the visibility of other members' work in a group activity. The increased visibility of the output boosts attention and invites other students to the action [Horn and Jacob, 2007a].

- **Accessibility:** Tangible tools can be used by students of all ages and abilities, making them more accessible than some other programming tools that may be more complex or require a higher level of technical proficiency. The intuitiveness of physical interaction lowers the threshold for children's participation. Supporting hands-on interaction increases the accessibility for both mental and physical disabilities.

Considering these benefits, tangible programming has proven to be a valuable teaching tool in K-12 schools, helping students to develop an understanding of programming concepts and build important problem-solving and computational thinking skills. Although using TUIs provides several advantages, some drawbacks of these interfaces limits their scalability. The previous research products were less scalable compared to text-based programming languages, as they may be less able to handle large amounts of data or handle more complex programming tasks. These tools focused on domain-specific instructions, which limited functionality compared to text-based programming languages, which can make it harder for students to create more complex programs. In addition, all tangible programming tools require using physical manipulatives that can be lost or damaged.

2.3 Designing Products for Children with Children

Researchers employ various methods, ranging from low- to high-fidelity prototyping techniques, to evaluate their concepts and gather input and preferences from children before creating final solutions. Sketching frames, paper prototyping, and similar cost-effective prototyping techniques have become standard practices in application development. In cases where the development process is expensive and requires unconventional prototyping methods, researchers turn to Wizard of Oz (WoZ) Prototyping [Maulsby et al., 1993]. This method enables researchers to simulate the intended behavior of the final product using affordable tools and technologies. In this setup, a human "wizard" assumes control over the system's intelligence and simulates interaction through a simulated interface. From sketching to WoZ Prototyping, all these techniques aid designers in making human-centered decisions. However, they still remain susceptible to designer bias since the designer oversees the entire process. By incorporating user-friendly AI development tools and involving children in the design process, we can enhance the product design process by mitigating designer bias and amplifying the voices, behaviors, needs, and habits of children.

Children can partake various roles in the design process. Druin identifies four roles in which children can actively participate: user, tester, informant, and design partner [Druin, 2002b]. In the user role, children engage with the technology while adults may observe, document, or evaluate their developed skills. Researchers leverage this role to comprehend the impact of existing technologies on child users, shaping future technologies and improving usability before production. As testers, children evaluate prototypes of unreleased technology. Through observation and direct feedback, their experiences influence the subsequent development stages of technology. In the informant role, children contribute to the design process at different stages, based on when researchers believe their input can inform design decisions. Lastly, in the design partner role, children are involved in activities such as interacting with existing technologies, providing input on design sketches, or assessing low-tech prototypes. Even after the technology is developed, children can continue to offer design ideas and feedback. By assuming the role of design partners, children are recognized as equal stakeholders in the design of new technologies throughout the entire development process.

2.4 Classroom Engagement: A Multi-modal and Multi-component Phenomena

One of the main goals of developing a tangible programming environment is fostering classroom engagement. *Classroom engagement* refers to the level of involvement and participation of students in an educational setting [Fredricks et al., 2004]. It is an important aspect of effective teaching and learning, as it helps to create a positive and interactive learning environment and can lead to better student outcomes. Classroom engagement can be fostered by integrating interactive teaching methods, group work, and hands-on activities. In our previous studies, we observed that sharing the programming cards and actively discussing the programming task increased the engagement of students in the learning process. The engagement also increases by asking and answering questions, providing feedback, and encouraging student-led discussions.

It is important for teachers to regularly assess the level of classroom engagement and to take steps to address any issues that may be preventing students from fully participating in the learning process. By creating a supportive and inclusive classroom environment, teachers can help students feel motivated and engaged in their learning. Yet, assessing engagement is a challenging task, as it requires evaluating multiple aspects of students' participation, especially for novice teachers. The classroom engagement of students can be evaluated by the following manual methods:

- **Observations:** A common method to assess classroom engagement is to observe the students during class using standard inventories [Marzano, 2007, Wang et al., 2014]. Looking for signs of engagement, such as active listening, participation in discussions, and involvement in activities, can reveal engagement.

- **Student feedback:** Asking students for their feedback on the level of engagement in the class using surveys or one-on-one conversations can reveal students' overall engagement [Black and Wiliam, 1998]. Yet, obtaining the precise level of engagement requires students to understand the concept and evaluate themselves objectively, which is generally a challenging task [Gao et al., 2021].

- **Checking learning and attendance statistics** Another way to assess engagement is to look at student grades, and assessments [Hattie and Timperley, 2007]. If students consistently perform well on assignments and assessments, it may indicate that they are engaged and motivated in class. Regular attendance can also be an indicator of engagement [Kuh, 2008], as students who are interested and motivated in the class are more likely to attend regularly.

2.5 Automatic Analysis of Classroom Engagement

Automatically analyzing classroom engagement is a growing interest in the educational technology research area. Although the automatic analysis would be limited to only those aspects of engagement that can be observed through video, using video recordings of students, we can inform teachers about engagement indicators. These observable features might include facial expressions, body language, and other non-verbal cues, as well as verbal interactions such as participation in discussions and responses to questions. The previous work in deep learning research demonstrated that current techniques could be utilized to understand engagement from video data.

Classroom Engagement Datasets

Existing work on engagement recognition presents only a limited amount of audio-visual dataset, which restricts the accurate representation of an engaging classroom environment. The need for a comprehensive and publicly-available dataset is acknowledged, yet the effort in this area is inadequate [Dewan et al., 2019]. Most of the video datasets have been collected in an online setting, where students interact with an interface while a camera faces directly toward the participants [Delgado et al., 2021, Yu et al., 2021]. Datasets like DAiSEE [Gupta et al., 2022] only include facial areas with limited affective state annotation. Although this kind of setup shows its benefits in online learning, a classroom environment is more complex and engagement measurement requires a more comprehensive analysis rather than looking at facial expressions. Sümer et al. collected audiovisual recordings of a traditional classroom in middle and high school students with a focus on engagement [Sümer et al., 2021]. Although this dataset considers classroom interaction compared to other datasets, their face recordings are too small to capture facial features, and the dataset is not publicly available.

Inventories for Assessing Engagement

To structure the list of these observable features, we utilized Wang et al.'s Classroom Engagement Inventory (CEI) [Wang et al., 2014]. The inventory is a tool for assessing student engagement in the classroom. Overall, the CEI provides a framework for understanding and assessing student engagement in the classroom and can be used to inform teaching practices and support student learning. It includes three dimensions: cognitive engagement, affective engagement, and behavioral engagement:

- *Affective engagement* refers to students' emotional involvement and motivation in the learning process. This can involve factors such as interest, enjoyment, and sense of belonging in the classroom.
- *Behavioral engagement* refers to students' participation in class activities. This can involve activities such as asking and answering questions, participating in discussions and completing assignments.
- *Cognitive engagement* refers to the degree to which students are mentally involved in the learning process. This can involve activities such as asking and answering questions, reflecting on and processing new information and solving problems.

Existing Video Analysis Models

While the number of datasets and machine learning models for classroom engagement detection is limited, the field of affective computing aims to address similar challenges. For example, one task in affective computing is predicting the valence and arousal levels of movie frames, which is similar in terms of input and output modalities to classroom engagement detection. Various supervised learning methods have been used to tackle this task, including Hidden Markov Models (HMMs) [Malandrakis et al., 2011], Convolutional Neural Networks (CNNs) with Recurrent Neural Networks (RNNs) [Nicolaou et al., 2011], and combinations of different deep learning models [Kahou et al., 2013]. These approaches require different representations of valence and arousal in defining emotions. For instance, as HMMs can only output discrete values, Malandrakis et al. predict seven discrete levels of arousal and valence. Meanwhile, LSTMs and other deep learning models can be trained to predict continuous or discrete variables, depending on the design of the model.

Additionally, we can borrow valuable approaches from the video action recognition literature. In the field of video action recognition, the task is to identify the action being performed in a video clip. This task is similar to classroom engagement detection and affective computing, as it involves processing video data to extract relevant features and make predictions based on these features. Various approaches have been proposed to tackle this task, including both traditional computer vision methods and deep learning-based approaches.

Traditional methods such as HOG and HOF have been used to extract features from video frames, which are then fed into a classifier such as SVM for action recognition [Dalal and Triggs, 2005]. More recently, deep learning-based methods such as Convolutional Neural Networks (CNNs) have achieved state-of-the-art performance on this task [Simonyan and Zisserman, 2014, Carreira and Zisserman, 2018]. Two-stream CNNs [Simonyan and Zisserman, 2014] and 3D CNNs [Carreira and Zisserman, 2018] have been shown to be particularly effective for capturing both spatial and temporal features from videos.

Recent research has also focused on developing efficient deep learning models for video action recognition, such as the Mobile Video Networks (MoViNets) [Kondratyuk et al., 2021]. MoViNets are designed to be computationally efficient while maintaining high accuracy, making them well-suited for resource-constrained applications such as mobile devices or embedded systems. MoViNets are based on a combination of depth-wise separable convolutions and inverted residuals, and they have been shown to achieve state-of-the-art performance on several video action recognition datasets while being more efficient than other deep learning models.

Other approaches to video action recognition include using optical flow [Wang and Schmid, 2013], attention mechanisms [Yue-Hei Ng et al., 2015], and graph convolutional networks [Yan et al., 2018]. Each of these approaches has its strengths and weaknesses, and the choice of approach will depend on the specific task and dataset being used.

Overall, video action recognition provides a rich set of techniques and approaches that can be applied to classroom engagement detection and affective computing tasks, given the similarities in the input modalities and feature extraction requirements. MoViNets represent a recent advancement in this field, offering efficient and accurate models that can be applied to a variety of video action recognition tasks.

Facial Landmarks and Action Units

Facial expressions are one of the most immediate notifiers of engagement [Whitehill et al., 2014]. Recognizing engagement via facial expressions is a long-time research interest in the affective computing area. One of the most successful and common methods to recognize facial expressions is selecting the action units (AU) for a multistate analysis of face [Tian et al., 2001]. Tian et al.'s expression analysis system use forty-four facial action units to describe the facial expressions in a single or additive fashion. For example, using this system, an action unit, AU4, means brows are drawn together and lowered, but when combined with another action unit, AU1, brows are drawn together but raised. Combining these action units can describe the affective states of students, which can be a key indicator of affective engagement[Dewan et al., 2019]. For example, the combination of changes in AU1, AU4, AU7, and AU12 can indicate confusion in the learning setting [Bosch et al., 2016].

Recognizing AUs depends on the model's ability to capture facial representation and find the accurate activations for each action unit. Considering the BP4D dataset evaluation on F1 scores, Swin-B-based Graph AU Detection Network [Luo et al., 2022] and OpenFace [Baltrusaitis et al., 2015] promise near state-of-the-art recognition results. This dataset consists of forty-one participants and annotations for eleven AUs [Zhang et al., 2014]. In our work, we utilized OpenFace's Action Unit Recognition on Multi-Person Videos.

Pose Features

The human pose estimation task aims to detect the poses of human body parts in 2D or 3D positions. RMPE (AlphaPose) [Fang et al., 2018], GPN [Nie et al., 2017], SPM [Nie et al., 2019], and OpenPose [Cao et al., 2019] models are some models that demonstrate state-of-the-art recognition results in the MPII-Multi-Person evaluation task [Andriluka et al., 2014]. OpenPose is an open-source system for multi-person 2D pose detection in near real-time [Cao et al., 2019]. The detection results in skeletal joint positions, including body, foot, hand, and facial keypoints. The system also presents near-state-of-the-art results in multi-person pose estimation results [Song et al., 2021]. Our work utilizes OpenPose to get insights based on students' actions and interactions with peers.

Interpretable Machine Learning

Interpretability in ML is an ongoing discussion that researchers consider model properties such as if the model is linear, presents a monotonous structure or allows interaction in the local-level [Molnar, 2022]. Logistic regression (LR) and k-nearest neighbor (kNN) are interpretable in terms of feature explainability. LR, in addition, is monotone, which ensures the parallelity between a feature and the target outcome directions. In this case, a change in the direction of the feature will cause the same direction in the outcome. But, this advantage also causes LR models to fail when the features and outcome are nonlinear, or features interact with each other. On the other hand, kNN is not linear or monotonous, but in small feature spaces, it allows us to find explanations to feature behaviors.

Two methods in explainable AI (XAI) research have been gaining weight: Creating glass-box algorithms that can result in transparent rules by definition and creating tools that can explain the behavior of the black-box closed models [Hagras, 2018]. Explaining the mechanism of black-box machine learning has no de facto standard. The Explainable AI (XAI) community actively investigates the possible methods for achieving effective communication of explaining and interpreting the machine learning models. One recent method that has achieved widespread adoption is Local Interpretable Model-agnostic Explanations (LIME) [Ribeiro et al., 2016]. The algorithm can explain the predictions from regressor and classifier models by approximating the local behaviors using pre-defined interpretable models. We utilized the InterpretML toolbox to call the LIME model after the training step to analyze and interpret the explainability rules [Nori et al., 2019].

Chapter 3

EXPLORATORY DESIGN STUDIES

Notes to Practitioners:

- I shared the workshop materials and guidelines on `https://karton.ku.edu.tr/atolye` (Turkish) and `https://karton.ku.edu.tr/workshops` (English)
- The workshop materials can be used to introduce what programming is and how tangible programming works. I observed that developing a programming language together with children helped students to understand the capabilities of computers and AI.
- At the end of the workshop, I curated a list of practical design considerations to guide students and teachers to use tangible programming materials in their programming activities in Section 3.5.2.

Related Publications:

- Exploring Children's Use of Self-Made Tangibles in Programming - `https://arxiv.org/abs/2210.06258`
- Prototyping Products using Web-based AI Tools: Designing a Tangible Programming Environment with Children `https://dl.acm.org/doi/abs/10.1145/3535227.3535239`

The classroom activity scenarios that involve using an intelligent tangible programming environment requires students to actively explore improved ways of framing their algorithms and considering the relationship between commands, inputs, and outputs via self-made tangibles while collaborating with their friends in this screen-free environment. However, integrating self-made tangibles to define abstractions in these paper programming environments brings two main challenges:

- Is representing abstract programming definitions by self-made tangibles educationally and pedagogically appropriate?
- Can self-made tangibles become well-designed code elements that can be memorized and understood by both makers and users of these tangibles in later programming sessions?

Before starting to develop an intelligent tangible programming environment, I conducted four-step "exploratory" workshop series with children. The term "exploratory" emphasizes the iterative and creative process of understanding children's needs, habits and abilities in self-made tangible creation. Later, following reflective thematic analysis [Braun and Clarke, 2019], I develop design considerations for tangible programming. Throughout the process, I conducted four studies as summarized in Figure 3.1:

1. In the first step, together with 7-11-year-old children, I decomposed a problem.
2. Next, children designed a set of tangible objects for programming the selected task.
3. Then, using a design probe that I specifically designed for these studies, I tested the memorability and understandability of children's self-made tangibles.
4. Finally, I tested the usability of a few-shot classifier for self-made tangibles using an already-available computer-vision prototyping platform, Teachable Machine.

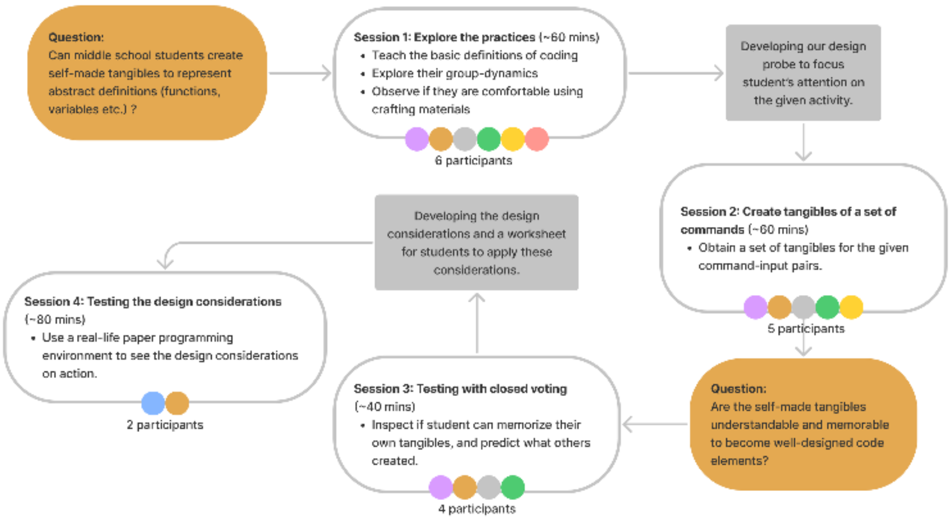


Figure 3.1: **The overall flow of the exploratory design activities:** Each activity builds upon each other and aims to answer a specific design-related question.

Throughout these design studies, children are involved in different roles as design partners. As I elaborated more detail in Section 2.3, children can take four main roles during the design and development process: user, tester, informant, and partner. In these exploratory design studies, I aimed to maximize the involvement of children. Involving them as real partners in a subject they have very limited knowledge of is a challenging design problem. But, I managed to include children in the design studies by gradually increasing the level of involvement and designing age-appropriate design probes for a systematic self-made tangible programming language design.

In this chapter, I describe four studies in which I gradually increased the level of children's involvement, the thematic analysis process, and design consideration list for producing self-made tangibles in programming activities.

(a) (b)

Figure 3.2: **The first session of exploratory design studies:** (a) Workshop setting. (b) Session 1 outputs. From left to right, student-made tangibles for potting flowers, calling the robot, walking around the garden, planting trees, mowing grass, and watering flowers

3.1 Session 1: Introducing The Concepts And Learning The Needs

Six children from Sariyer, Istanbul (one seven-year-old, three ten-year-old, and two eleven-year-old children) participated in the first study. First, I introduced programming and discussed its role in many breakthroughs in society. Later, I asked them to define a problem or a task to solve with programming. The children came up with the task of "gardening" and started decomposing the possible sub-tasks that could be solved with programming. After listing all the tasks, we agreed upon eight sub-tasks: potting flowers, planting trees, protecting the garden, watering the garden, mowing grass, removing weeds, walking around, and picking fruits.

In this workshop, the aim was to see if children were comfortable with the given materials and their ability to understand the task at hand. The observations revealed that all students were comfortable with the provided materials and were eager to explore programming concepts regardless of their age. As seen in Figure 3.2(b), almost all students created tangible objects with different materials. The sizes of the tangibles were similar in each child's creation. Overall, this showed that the presented materials were diverse enough to accommodate children's preferences.

Throughout the workshop, I observed some tendency to disengage from the task due to the informality of the workshop. To explain, some children knew each other and also the vicinity. While the familiarity led them to collaborate easily on the given task, it quickly turned into a free chat and distracted them from the task. It is known that group dynamics directly impact children's creativity [Van Mechelen et al., 2014]. Based on this experience, I handed out a design probe (seen in Figure 3.3) for the next workshop to help focus children's activities.

3.2 Session 2: Designing Tangibles Of A Set Of Real-Life Commands

Five children from the previous study (one seven-year-old, two ten-year-old, and two eleven-year-old) participated to second study; one child could not attend the workshop. In this session, I distributed the design probes to help them maintain their focus. I hung a larger copy with commands and possible input names to a wall that all students could see. I handed out empty copies of this probe and asked them to place their final self-made tangibles on pre-defined positions. As intended, the probe helped them to use their time and communicate their ideas more efficiently.

In this session, students created a programming interface for gardening with a total of eight commands, as decided upon earlier. I asked children to design these tangible representations so that a computer (a simple electronic machine) could clearly understand them. Overall, I obtained forty different programming objects from five children. Figure 3.3 demonstrates two distinct samples of these probes that show different input modalities.

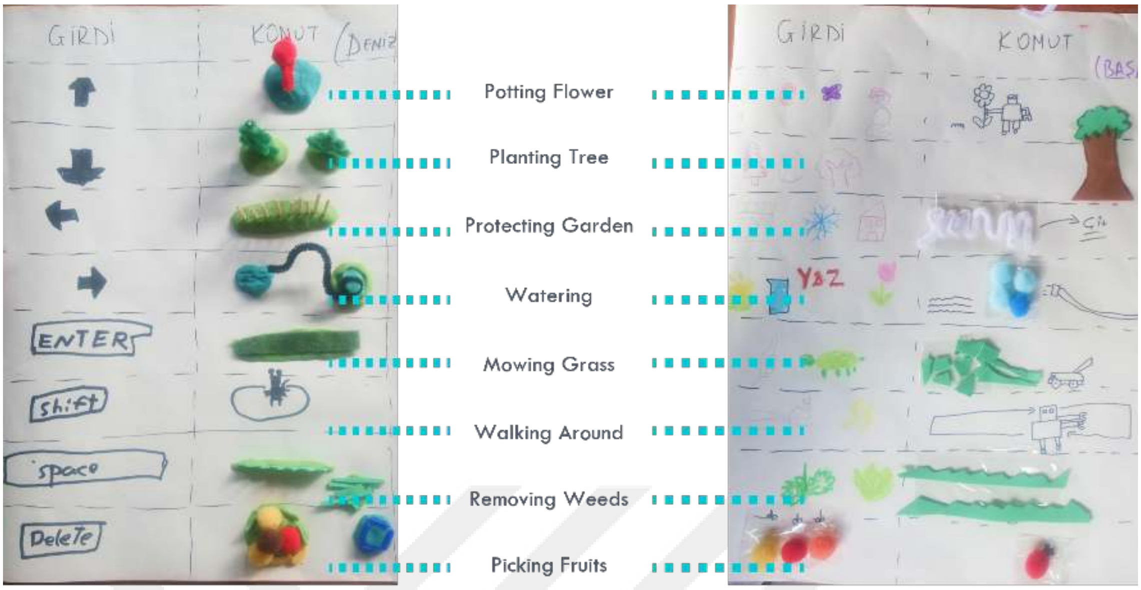


Figure 3.3: **Self-made tangibles from two participants**: (*Left*) The participant (he) offered to use keyboard presses to change the given input. He shows the command first, then selects the possible inputs using keyboard shortcuts. His friend also shared the same design decisions, which resulted in very similar command-input pairs as expected. (*Right*) The participant (she) used a combination of drawing and given materials to create programming objects. Providing input to these commands can take different forms. For example, changing the flower's color in the robot's hand is the input of the "potting flower" command, but watering the garden requires combining two separate drawings.

After collecting all the objects and seeing how children created these tangibles, I asked if these objects were qualified to become coding elements. A well-designed code element should be easily understood by others and easily memorable by the author of the program. To this end, I tested the self- and inter-memorability and understandability of the programming objects. In this context, I defined understandability as the ability to correctly interpret the meaning of the given tangible objects by other students. And memorability is the ability to remember the represented role of their own tangible object when they see the tangible object later.

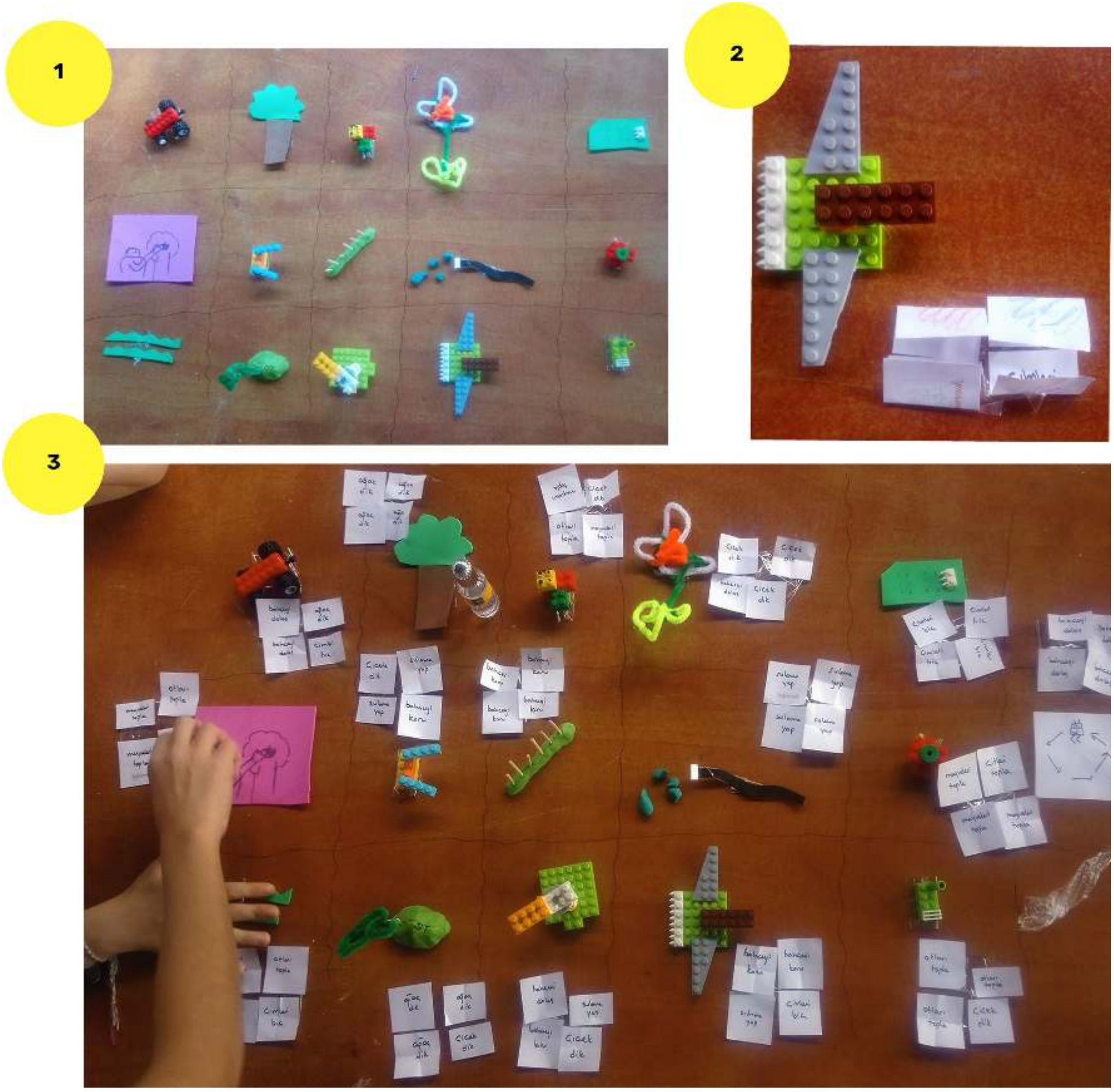


Figure 3.4: **The third session:** Before children arrived at the workshop area, I prepared a grid of programming objects on a table. Children placed color-coded function names to see if they remember their own self-made tangibles and understand others.

3.3 Session 3: Testing The Memorability And Understandability

Four children from the previous studies (two ten-year-old and two eleven-year-old) participated in this study. Before the children arrived at the workshop area, I prepared a grid of programming objects on a table. I handed color-coded function names to children to test the understandability and memorability of the workshop's outcome.

1. Sixteen blocks (four tangible blocks from each participant) were selected and placed on a table.

2. I distributed color-coded small paper pieces and handed them to the children. Each child is assigned one color to keep track of recognizing their tangible objects, and they place the enfolded paper pieces onto the grids. They placed this paper onto the grids in a closed form.

3. I discussed whether they remembered these commands or predicted their friends' tangible outputs. I tried to understand the effect of using different materials and different representation methods.

Figure 3.5 summarizes the test results by showing the understandability accuracy (top-right corner of each grid), the correct label of the tangible object (bottom-left corner of each square), and each child's answer with color-code information (bottom-right corner of each square). These results indicate that three main patterns affect the understandability and memorability of a programming object in general:

1. Using materials that drive children to create abstract shapes (construction bricks) reduces future understandability.

2. The object's recognizability also increases when children depict the action stated in the command rather than directly imitating the object's appearance. Trying to tell the action increases the level of detail.

3. Similarly, using more than one material increases the level of detail.

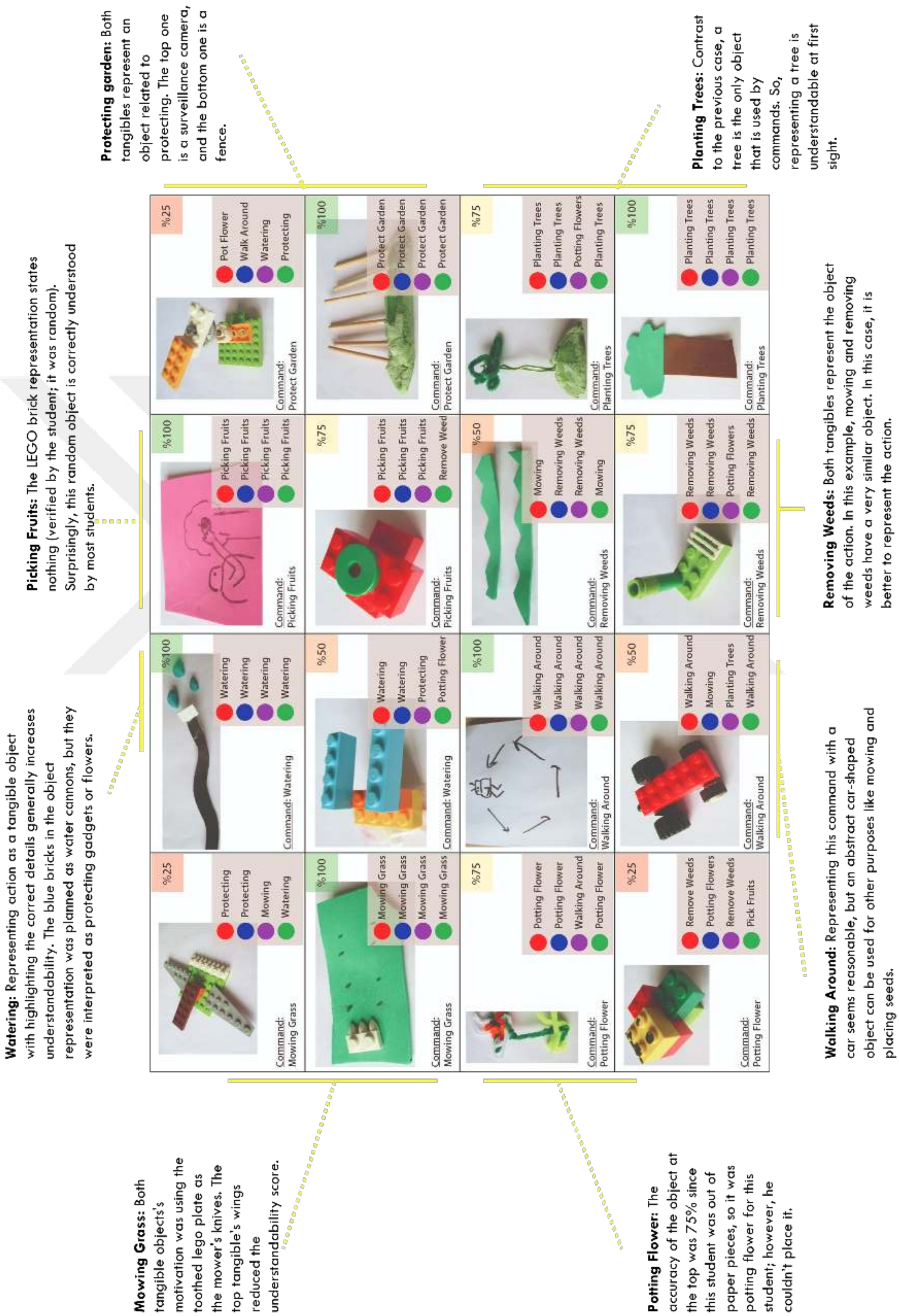


Figure 3.5: **The understandability and memorability results**: The understandability scores of tangible objects. Each command has two different representation that uses different materials.

3.4 Session 4: Introducing Child-Made Tangibles to a Computer Vision Powered System

In this activity, participants were engaged in creating a novel high-level programming interface utilizing tangible blocks for the Flappy Bird Game. The entire study lasted for approximately one and a half hours. Initially, I provided an overview of the study process and instructed participants on the fundamentals of sequential program structure and the utilization of events in programming. Subsequently, I broke down the actions involved in the Flappy Bird game and introduced ten programming commands that could be employed to code the game from scratch. For each command, participants designed tangible representations using various craft materials, such as play dough, wires, crayons, and so on. Following that, they utilized Teachable Machine to construct the recognizer model and used the model's shareable link in the game-development interface.

(a) (b) (c)

Figure 3.6: **Self-made tangibles for programming Flappy Bird game:** Figure shows three participants' tangible representations of Flappy Bird game's programming commands. In the activity, they had fifteen minutes to build ten tangible representations. (a) This participant used cardboard and playdough, (b) She drew the actions/objects on different colored cardboard (the color choice was random) (c) He used metal wires and pipe cleaners to represent objects.

3.4.1 Coding the Game with Tangible Representations

In the game coding interface, children show their self-made tangibles to add events and commands in the correct order to code the game. Ten pre-defined functions allow children to create a Flappy Bird game from scratch. These commands are "When Touch Screen," "When Game Starts," "When Pass Obstacle," "When Hit Obstacle," "When Hit Ground," "Flap," "Change Score," "Set Speed," "Place Obstacle," and "End Game." I asked them to build new tangible representations for these predefined events and action commands. I also handed out a probe with a grid table showing all ten pre-defined functions to help children track their creations. Children had only fifteen minutes in this session, which led them to use much simpler materials like paper, pen, and play-dough, as seen in Figure 3.6

3.4.2 Using Teachable Machine

After creating the physical correspondences of these functions, children used the *Teachable Machine* as seen in Figure 3.7. Opening a new Image Project in Teachable Machine shows an intuitive interface where children encounter a traditional machine learning pipeline. The application interface serves a traditional machine learning pipeline with three steps: (1) Users collect data by either uploading images or capturing them in real time via webcam, (2) Once the data is ready for all class labels, users can start training with pre-defined hyper-parameters. Experienced users can tweak the parameters to increase speed or accuracy. (3) Finally, users can test the results in real-time. If the user is happy with the results, they can export the model using a single link. Alternatively, they can collect more samples or change the parameters and fine-tune their models to improve the classification accuracy.

In all studies, children either needed to update their model by adding new samples to the dataset or changing the tangible objects' design. When the AI model was not working as expected, I stopped and asked what can be the model's problem. They neither blame the system nor assume the system is broken. Some common guesses and suggestions were increasing the number of samples, using better lighting while taking photos, and using diverse samples that the system can easily differentiate. These suggestions indicate that children can use and improve AI systems with sufficient guidance. After the discussion, I guided them by adding more samples or updating the tangibles to separate visually similar objects better. Utilizing Teachable Machine in the design process allowed us to see the tangible programming objects in action, and children felt their impact on the design process.

Figure 3.6 shows sample tangible programming blocks that issue different challenges. (a) One child used yellow cardboard to sketch figures. When he created three visually similar tangibles, I asked him to test the model. As expected, the confidence score of the test results was not reliable. Before explaining why the test resulted in this, he already made the deduction. Then, he changed the tangibles with different materials like play dough and gave them different shapes. (b) In this set, the child sketched or wrote some text on the cardboard. With an analogy to application buttons, the pink cardboards have some text on them. Teachable Machine can recognize different text when the training data is rich, and sample images are close to the camera. When the model is trained this way, I always needed to show these cards very close to the camera and shoot many images from different angles, but it was not a problem for her. (c) This child picked metal wires in the tangible building phases. For example, the "When Touch Screen" command is represented by a wire hand. The problem is that metal wire exposes all the background, and the AI model mostly learns the background image features. Additionally, metal wires' visibility quickly changes in different light conditions. After seeing the confidence scores of the AI model, this child chose a different material.

3.5 Design Considerations for Self-Made Tangibles

After conducting four sessions, the observations and analysis of tangible outputs led us to curate design considerations for building self-made tangibles. Although testing the setup with a small-scale group with participants in different age groups restricted the generalizability, the observations and analysis led us to develop early design considerations.

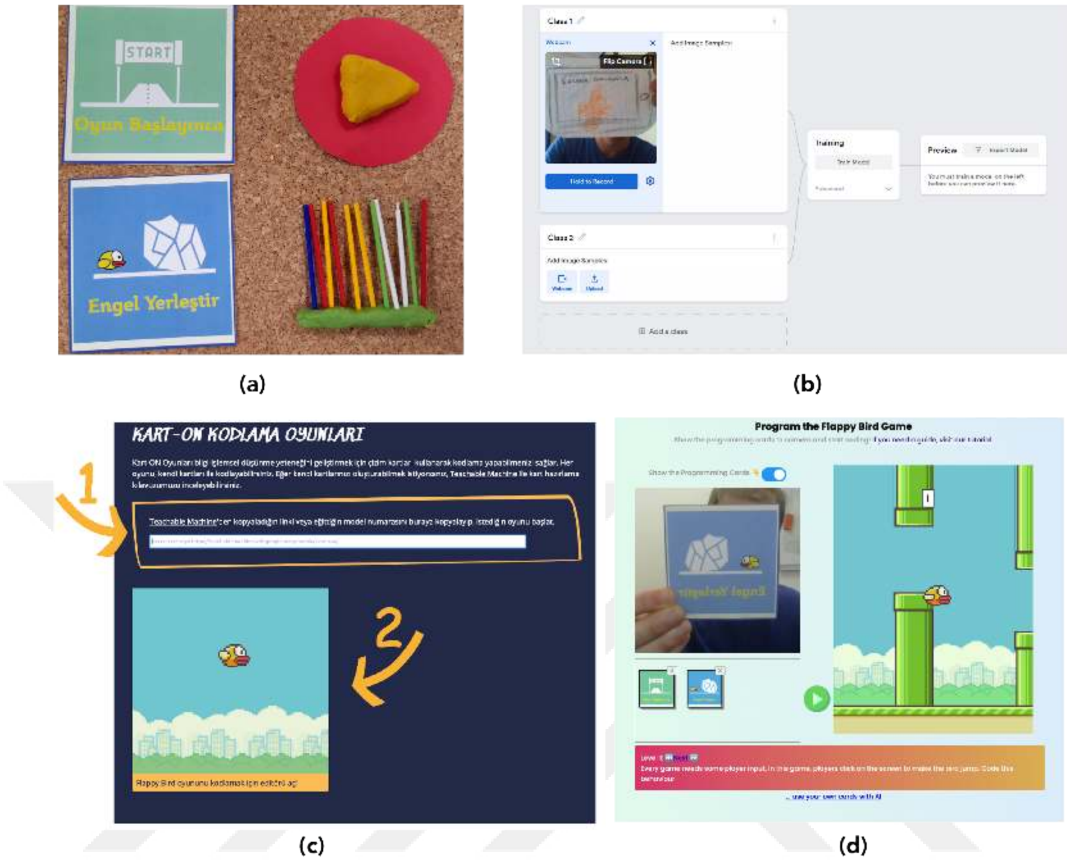


Figure 3.7: **Programming flow by introducing self-made tangibles**: (a) Children created tangible representations of pre-defined commands. (b) They used the Teachable Machine to create the computer vision model. Children collect data via webcam, hit training when the data collection is completed, and test the recognition accuracy. (c) They entered the AI model link to use it in the game. (d) They used their tangible programming blocks in Flappy Bird coding interface. First, they show the tangible representation to the webcam. If the model recognizes the representation, it appears in the coding area. Then, they click the "play" button, which runs the game with the coded logic. This frame shows the output of two programming cards: "When the game Starts" - "Put obstacles," which results in a game interface with obstacles.

3.5.1 Observations and Analysis

In the analysis, I used my observation notes and tangible output photos. I thematically categorized the observations into three categories: Materials, Improving the vision model and Interaction with the AI system.

- **Materials:** Five students created a total of forty tangible function representations, and fifteen of these tangibles were completely created with LEGO's. Although students enjoyed using these, the understandability and memorability results are considerably low, compared to other materials. Using play dough, wires and other shape-changing materials is great for adding details, but not long lasting, which decreases the repetitive use of self-made tangibles as an algorithmic structure.

- **Improving the vision model:** Children created objects from wires and other transparent materials. The computer vision model could not differentiate these objects as using these materials shows a larger portion of the background. So, this type of material selection that increases transparency reduces the accuracy of the vision model.

  If children use the vision model in different backgrounds, or if the light conditions of the setup change rapidly, taking photos in different backgrounds or adding data augmentation by applying a segmentation beforehand is required.

  Considering different environmental conditions is important as environment lighting in the case of computer vision, and noise in speech systems determines the accuracy.

- **Interaction with AI system:** Children above the age of seven (N=4) quickly learned the relation between model accuracy and the number of data samples. They easily grasped if the model needed more data to improve the model, yet they needed an intuitive method of adding more data to the right label.

  Conducting the activity with a seven-year-old participant allowed us to get a glimpse into younger age groups' use of the Teachable Machine for prototyping. During the study, he needed constant support from the main researcher, but eventually was able to introduce new tangible representations for a new class.

3.5.2 Design Considerations for Building Self-Made Tangibles

These exploratory prototyping processes allowed us to learn the challenges before developing the computer vision model quickly. I listed four main design considerations (DC) based on the observation notes, material analysis of students' tangible creations, and memorability/understandability test results. These considerations aim to enrich semantic layers in the building phase of tangible objects to improve the object's understandability and memorability. I also considered the fact that these models would be used in computer vision systems in real-life programming applications.

- **Give hints about the action:** The most understandable and memorable designs involved representing the object's visual resemblance and stated action together. For example, "watering a garden" can just be represented using a "hose." But, in the test, I observed that students can interpret this as a protection object and say the command is "protecting the garden." So, adding contextual layers and giving hints about the stated action supports the understandability of the self-made tangible.

- **Choose the right materials for the environment:** Considering the materials for different locations was an important aspect of the quality of programming time. If the students plan to use the objects more than one time, using play-dough is not encouraged since it cracks after drying. Also, considering the fact that these materials will be recognized by a computer vision system in future studies, light conditions can affect the vision model's accuracy significantly. Similarly, using only wires to represent programming objects result in a high dependency on foreground- background clarity. Therefore, the environment should be considered while choosing the materials.

- **Combine materials:** Combination of materials enhances the interpretability of programming objects from the child's perspective. Using combinations of different textures and colors also increases the accuracy of the AI model's recognition.

- **Add details with 'simple' shapes:** Adding details helped children to memorize the function of designed tangible and enhanced the AI model accuracy. Yet, I emphasized adding details with 'simple shapes' to use children's time more efficiently. In the tangible creation process, they tended to add details with 'fancy' materials and zigzagged scissor moves, which distracted them from the tasks.

3.6 Implications

Activities that encourage children to design self-made tangibles of abstract definitions can be useful in engaging students and promoting active learning. Throughout the studies, the observations demonstrated that asking students to create tangible representations helps them understand the code, which resonates with the research in active learning [Fyfe et al., 2014]. For example, using student-made tangibles in math learning for young children can help them engage in activities, familiarize the concepts in real-world situations and help them understand relational information better. Additionally, using self-made tangibles helped students focus on a code scope's overall working mechanism.

On the other hand, students and teachers need guidance for tangibles to become usable in abstract definitions of programming elements, such as defining new variables and functions. The outputs created in Session 4 indicated that informing children about design considerations may not directly result in improving the quality of tangible objects. Although having a set of considerations helps workshop moderators to guide children more effectively, children can choose to create the tangibles with their favorite materials rather than the logical ones. So, rather than delivering these design considerations simultaneously, I suggest practitioners explore these considerations together with children to increase their understandability and memorability in creating tangible representations.

Using the web-based AI training tool *Teachable Machine* allowed children to create their physical products in real-life scenarios. In traditional co-design activities, children or adult users mostly build paper prototypes or other low-fidelity design outcomes. The studies demonstrated that using a functional development environment with novice users on an unfamiliar task increased the involvement of students in the design process and supported their iterative learning process.

Based on the qualitative observations and analysis, I can summarize the answers to the main research questions as follows,

1. *Can students in K3-6 grades build a tangible representation of abstract programming definitions?* Yes, I observed that students liked building tangible representations of abstract programming definitions, and they could link the function with the representation.

2. *Can students understand and memorize self-made tangible representations in later uses?* It is not an easy task, even with the guideline. The list of design considerations helped students manage their time and use the materials in a structured way, yet they still had challenges in following them.

3. *How can researchers support children in creating tangibles that can yield high-accuracy recognition results?* Following the design considerations, adding details about action, supporting diverse textures and colors, and adding details with simple shapes helps AI model to recognize the model and enhance the understandability and memorability of the programming objects.

Chapter 4

KART-ON PLATFORM FOR AFFORDABLE AND ENGAGING PROGRAMMING EDUCATION

Notes to Practitioners:

- The mobile application and TFLite Model implementation are open-source and available on `https://github.com/karton-project/karton-android`.
- I developed a set of helper tools that are not part of this thesis. For example, a web application that teachers can show Kart-ON blocks and their outputs in a drag-and-drop visual style to ease the introduction via smart boards. These tools are also open-source and available at `https://github.com/karton-project/kart-on-activities`.
- The programming style has proved itself valuable for promoting group work and increasing affordability. I developed a tangible music programming platform for visually impaired students' programming education (visit Budgie Project Webpage) and a tangible data programming platform (visit Martı Project Webpage) that applies the same principles with our main programming environment. These are the side research projects that I did not include in this thesis.

Related Publications:

- Kart-ON: Affordable Early Programming Education with Shared Smartphones and Easy-to-Find Materials - `https://dl.acm.org/doi/abs/10.1145/3379336.3381472`

- Kart-ON: An Extensible Paper Programming Strategy for Affordable Early Programming Education - `https://dl.acm.org/doi/abs/10.1145/3534524`

Kart-ON is an intelligent tangible programming environment that fosters collaborative and engaging programming activities for groups of students while reducing dependence on electronic computing devices. The programming environment comprises two main components: (1) Physical programming cards that can be arranged to create new code parts and (2) a mobile application that uses custom tangible representations to define functions. Unlike traditional programming environments that require students to work on personal computers, Kart-ON allows large groups to work on programming tasks together using a shared smartphone or tablet at a low cost. The environment also enables the use of custom tangible objects as programming elements, offering students the flexibility to save functions and other algorithmic structures by associating them with their favorite physical objects or using inexpensive everyday materials such as paper, clay, and markers.

To ensure the effectiveness of Kart-ON, I engaged in an iterative design process that involved testing usability, collaboration, and engagement with a diverse group of 116 students (9-12 years old) from various schools and NGOs, either individually or in a classroom setting. Figure 4.1 summarizes the flow of the studies conducted through the development of the Kart-ON environment. In these studies, I used our paper programming environments as an introductory tool to teach programming and observe students' behavior while using digital and tangible interfaces. Through these studies, I observed that the use of tangible blocks in programming activities enhanced collaboration and engagement while supporting students' focus and understanding of the task at hand. This research provides valuable insights for supporting equal opportunities in programming education and serves as a guide for other developers looking to design similar paper programming environments.

In this chapter, I provide a detailed account of the design and development process of Kart-ON's paper programming cards, the mobile application, and user studies. Based on UX study observations and analysis results, I offer a set of design considerations that can inform the creation of paper programming environments suitable for K5-8 grades (9-12 years old). Additionally, I explore the unique advantages and challenges presented by this new paper-programming paradigm for teaching programming, as revealed through user studies.

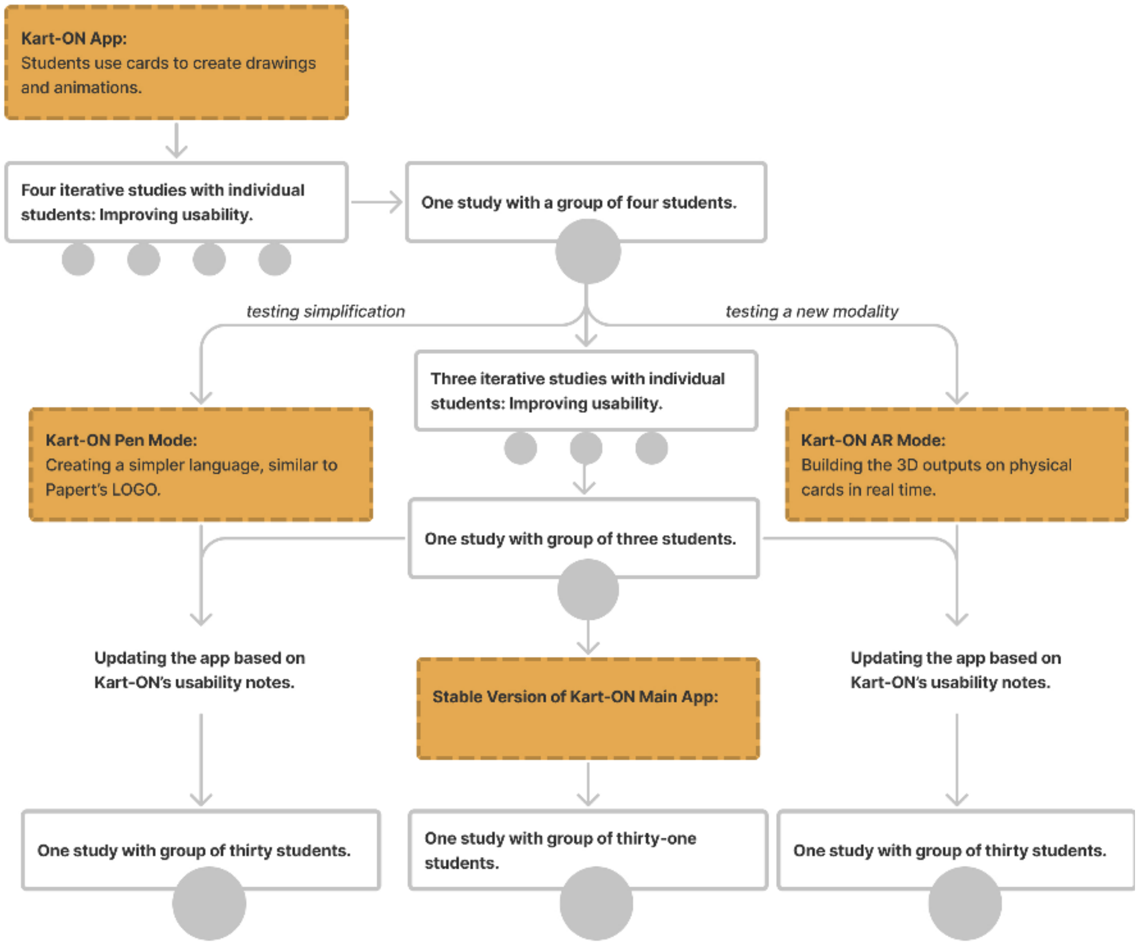


Figure 4.1: **The user study flow and corresponding changes in the product development**: In the end of the classroom studies, I synthesized design considerations to accelerate the adaptation of paper programming.

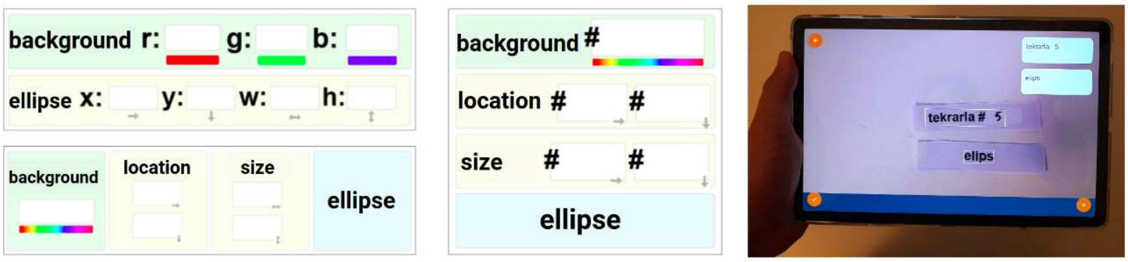
4.1 Kart-ON's Programming Environment

Kart-ON is a classroom tool that facilitates collaborative coding tasks using paper programming cards, allowing groups of 3 to 4 students to conduct creative coding activities. These tasks range from simple activities like drawing a snowman to more complex challenges like animating an airplane. Consider a scenario where the teacher assigns a task to animate a basic shape. This task involves defining a variable, incrementing its value in a loop, and utilizing it as the shape's coordinates. Each group may progress at a different pace, encountering challenges that require multiple attempts. The teacher can provide feedback on the code without scanning it, or they can scan the code, demonstrate errors to the students, and offer hints. While one group focuses on resolving an error, the teacher can move to another group and provide assistance.

Kart-ON utilizes the *p5.js JavaScript library* to create shapes and animations on smartphones. This library, a derivative of Processing, a framework designed to facilitate creative coding projects using computing devices, enables the development of innovative outputs. Creative coding, a field within computer science, harnesses computer graphics and other output modalities to give life to imaginative concepts. Its popularity has grown in educational settings, fueled by the burgeoning maker community and the emergence of user-friendly libraries such as *p5.js*. Teachers and curriculum developers have embraced creative coding frameworks, integrating them into computer science education at middle and high school levels. Scratch serves as another prominent example of such frameworks.

To design the mobile application and programming cards, I followed an iterative approach, testing multiple design decisions with children. Figure 4.2-a shows earlier versions, while Figure 4.2-b shows the current version of an example program that uses the programming cards to create an ellipse drawing with a background color. The quality of the session is determined by group dynamics, and in an ideal environment, one smartphone is sufficient for completing tasks in a classroom. However, providing a smartphone for each group and continuously testing the codes might be more suitable for another classroom.

In the rest of this chapter, I present the final version of Kart-ON by walking through the design decisions I arrived at through user studies.



(a) **(b)** **(c)**

Figure 4.2: **The last three iterations of card design**: (a) In the first iteration, input fields start with reminder initials with hint icons. I keep these icons for all iterations. In the second iteration, I tested horizontal ordering. (b) In the last and final iteration, I keep the vertically aligned commands. I limited the maximum number of input fields to two. (c) Students can show the programming commands to the mobile application one by one or multiple at once.

4.1.1 Programming Cards

Kart-ON has been designed to facilitate the quick creation of drawings and animations by students through its programming commands. These commands enable students to draw primitive shapes, define variables and functions, control program flow using *if/else* statements and *for loops*, add music, and perform mathematical operations. The programming command subset is highly expressive, being a subset of *p5.js* commands, which enables the creation of all control structures, including nested statements and callable states. Each control structure's scope is defined by the location of the *end* command card.

A Kart-ON command card consists of *command* and *input* fields. A command card includes a maximum of two input fields, and each input field starts with a hash symbol (#). The hash symbol is a visible reminder for students and also a distinguishable character for our computer vision model. The command cards are vertically stacked to create programs, and each card belongs to a category based on its functionality. These categories are primitive shapes (blue), color (green), size and position (yellow), control (red), and variables (pink), with each category having a distinct color to enhance card recognizability.

The design process for Kart-ON's programming cards was iterative and involved testing multiple card designs with children, as shown in Figure 4.2-a and b. In the first iteration, the pre-defined programming cards were an exact replica of Processing [Reas and Fry, 2007] commands. However, user studies showed that if a card contained more than two input fields, it took an excessive amount of time to introduce all the concepts related to these input fields, and students were often distracted. For instance, the background command in Figure 4.2-a had three input fields for defining color with R, G, B values, requiring students to learn the RGB color space to use the card. In the current design, students only use a hue value to use the card, learning the HSL color space through experimentation. Students can vertically stack the cards to create programs and can either scan each card and run the program to test its output or scan multiple cards at once and run them all. Although stacking the cards horizontally was also tested, most children preferred vertical ordering, as shown in Figure 4.2-a (bottom).

4.1.2 Mobile Application

The Kart-ON mobile application presents a selection of examples and learning content to help students build new programs without the presence of a teacher. They can create new programs and save them by scanning the command cards. Any Android smartphone (>v18, which covers 70.2% of all in-use smartphones and 99.3% of all Android OS devices) with a camera can run this application.

Coding with programming cards

The coding process for Kart-ON begins by using a camera to recognize programming cards. This is accomplished through the use of Google MLKit [Google MLKit, 2021] to detect the text on the cards. Students have the option to show the programming cards one at a time or all at once. The number of vertically stacked programming cards that can be used is determined by the size of the cards and the students' arm length. Once a physical block is recognized, it is displayed as a separate graphical block on the coding screen. Students are able to modify the commands and inputs through a graphical interface. Once the program is completed, the recognized text commands are interpreted into p5.js code. This code is then rendered and run on WebView. After scanning a card or stack of cards, the contents are displayed on the screen. To test the output of the program, students must manually click on the compile button. Once the code has been compiled, students can update the output by physically changing the cards or using the touchscreen.

Correcting the recognized text

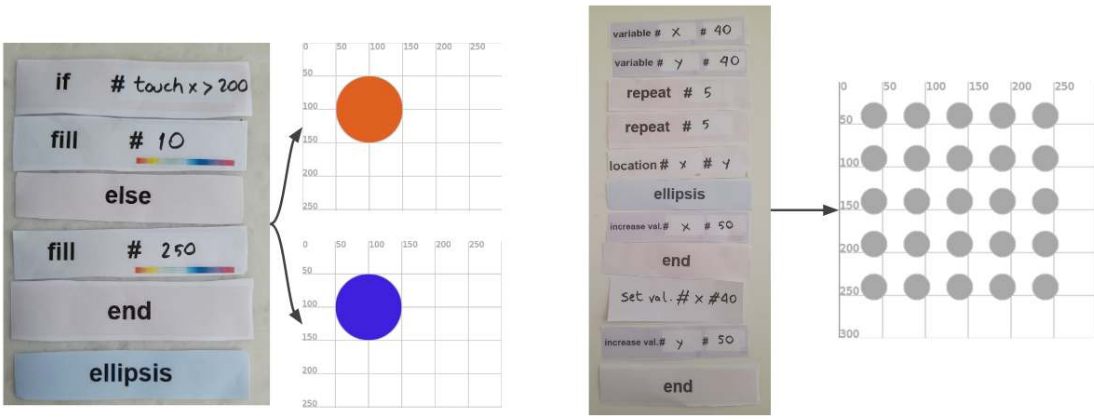
As the programming environment presents a limited set of commands, the application runs a fuzzy search to match the recognized text with the predefined strings. For instance, if the command "fill # 120" is recognized as "full # 120," the interpreter corrects the command automatically. I also use a fuzzy search to correct input fields when a variable is used. However, for numeric inputs, I provide easy-to-use sliders to change these inputs quickly.,

Saving functions by using physical objects

Students can save programs by assigning a name or a physical object via the camera. To accomplish this, I use a transfer learning approach similar to Senchanka et al. [Senchanka,]. Our pre-trained base model uses the MobileNet v2 architecture, which was trained with the ILSVRC-2012-CLS dataset. The transfer learning technique uses a model that is already trained with big data to retrain the last layers to solve a task that involves a few examples. When new examples are introduced, a fully connected layer with softmax activation is trained on top of this model. Students can assign physical objects to recall functions and programs by taking at least ten poses of these physical objects and training the model with these images.

Example Scenario:

Figure 4.3 shows two examples that uses a conditional and a loop structure. To run these applications on Kart-ON, students open the application, select the "Code" category from the navigation bar, tap on the "Create a New Program" button on the opened page. When they tap, they see the camera's view. In this step, they can show the cards one by one or multiple at once. To add the physical cards to the program, they tap the "Add New Block" button. In the case that the camera vision algorithm might give wrong results, students can tap on the block on the screen and change the command or inputs. When the students complete introducing all blocks to the application and are happy with their program, they tap the "See the Result" button.



(a) (b)

Figure 4.3: **Two example program definition using Kart-ON Drawing mode commands**: *a) Conditionals*: The program changes the ellipse color based on touch sensor position. If the horizontal touch position is more than 100 pixels, makes the ellipse red, else makes the ellipse blue. TouchX returns a numeric value between 0 and device-width and can also be used as an input in other commalnds. *b) Loops*: The program creates a nested loop to draw a 5x5 grid of ellipses. This program uses "x" and "y" variables to control the location of the ellipsis.

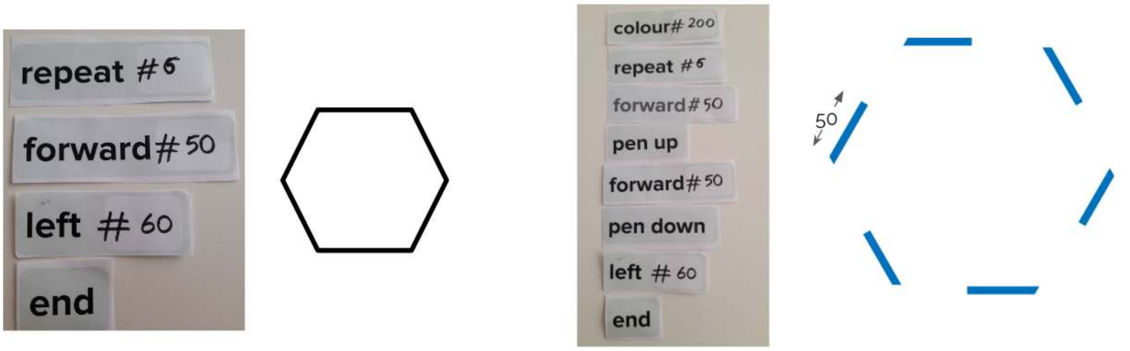
Figure 4.3 illustrates two examples that use a conditional and a loop structure. To run these applications on Kart-ON, students follow these steps: they open the application, select the "Code" category from the navigation bar, and tap the "Create a New Program" button on the next screen. This opens the camera view, which allows them to show one or multiple programming cards at a time. Students can add physical cards to the program by tapping the "Add New Block" button. In case the camera vision algorithm produces incorrect results, students can tap on the block on the screen and edit the command or inputs as needed. When the input is numeric, they can use easy-to-use sliders to adjust the values quickly. If the input is text, they can use the textbox. Once the students have introduced all the blocks to the application and are satisfied with their program, they can tap the "See the Result" button to run the program.

4.2 Extending the Capabilities of Paper Programming

I developed two extensions of our card-based programming strategies. First, I developed Pen Mode as a beginner tool, which follows a simpler coding style similar to the LOGO programming language. Second, I tested how different output modalities can use this card-based programming strategy and employ smartphones and developed a set of commands to create 3D shapes and animations in an AR environment.

4.2.1 Extending as a Starter Tool: Pen Mode

In the user studies, I also observed the need for a simpler tool that teachers can use to introduce the basic concepts about coding style, coordinate systems, and simple geometry. Traditionally, educational programming environments provide a simplified subset of a programming language with graphical elements for novice users to understand the grammar, syntax, and control elements. For example, Python's Turtle Graphics [Python Docs, 2021] and JAVA's Stanford Karel Robot [Pattis et al., 1981] allow users to solve puzzle-like maze questions to better understand the fundamentals of the language. In a similar fashion, the starter tool, Pen Mode, introduces a simplified version of Kart-ON. The tool borrowed its core idea from Papert's LOGO and its companion turtle robot, the first programming language to introduce computational concepts in primary and middle schools. Pen Mode also demonstrates an example case for our extensible paper programming strategy, where students can seamlessly switch between two programming environments. From a



(a) (b)

Figure 4.4: **Two example program definition using Kart-ON Pen mode commands**:(a) Drawing a hexagon with the Pen Mode Cards. (b) Drawing a dashed hexagon with the Pen Mode Cards.

technical perspective, Pen Mode uses the same mobile camera scanner and code parser with Kart-ON, which presents an example scenario for future developers to extend the Kart-ON's capabilities.

Students can switch between Kart-ON and Pen mode inside the mobile application. Similar to LOGO, the pen agent moves on a canvas and draws geometric shapes with given commands. For example, I can draw a simple hexagon by using *forward* and *left* commands as seen in Figure4.4-a. *forward* command draws a straight line starting with a specified length. *left* command will rotate the drawing agent's head towards the specified angle. Repeating *forward 50*, *left 60* commands one after the other six times will result in a hexagon. Figure4.4-b shows the programming cards to draw a dashed hexagon in Pen Mode. *pen up* and *pen down* blocks determine either the pen agent's color is visible or not. Moving forward with *pen up* state and then moving forward with *pen down* state will result in dashed lines.

4.2.2 Extending the Output Modality: Augmented Reality Outputs

Similar to Kart-ON's programming methodology, I developed an augmented reality (AR) version in an attribute-based language style. In this version, students scan the cards one by one to build 3D models and animations. The command set contains three groups of functions: Shape-creation, stylization, and animation. Shape cards include DRAW CUBE, DRAW SPHERE, DRAW CONE, and DRAW CYLINDER functions.

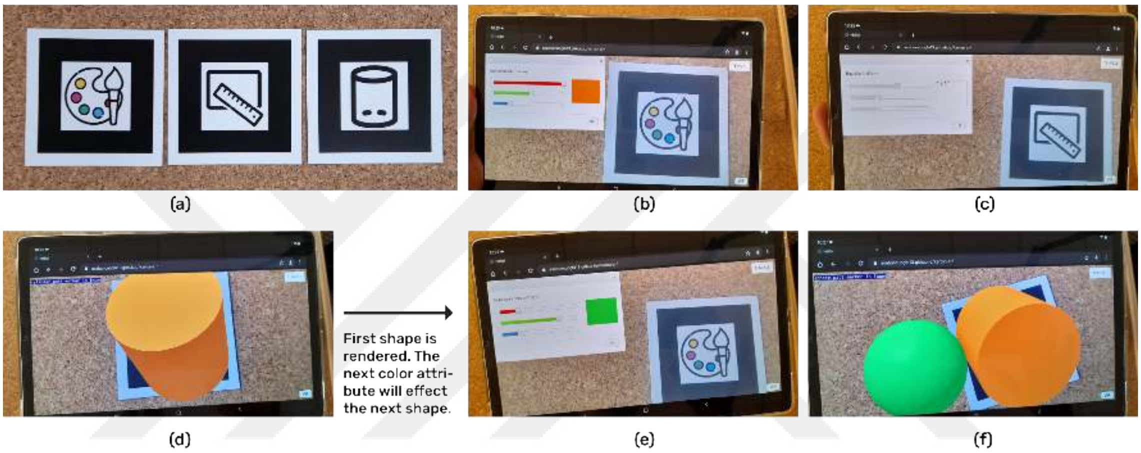


Figure 4.5: **An example program definition using Kart-ON AR mode commands**:(a) Three programming cards, from left to right, PICK COLOR, SET SIZE and DRAW CYLINDER. (b) A digital pop-up screen appears when the user shows the PICK COLOR marker to the AR camera. (c) Another digital pop-up screen appears when the user shows the "setting shape size" marker to the AR camera. (d) After determining the color and size via the digital interface, the user scans the sphere card. The output is rendered on the "draw a sphere" marker. (e) To add a shape with another color, the user needs to determine the color by scanning the color picker card. (e) This color is applied to the new shape on the sphere marker. As the figure shows, the new shape is rendered next to the previous shape automatically. The user can pick a location using the "setting shape position" marker to determine the position.

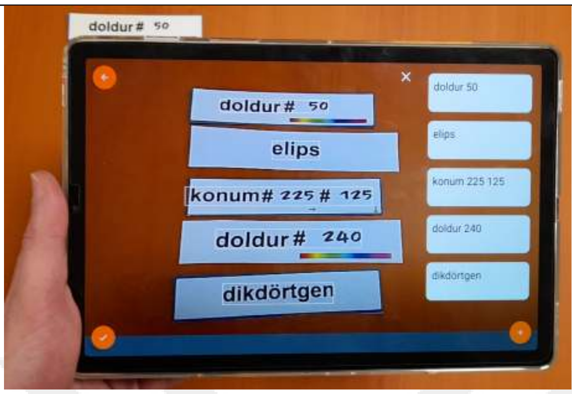
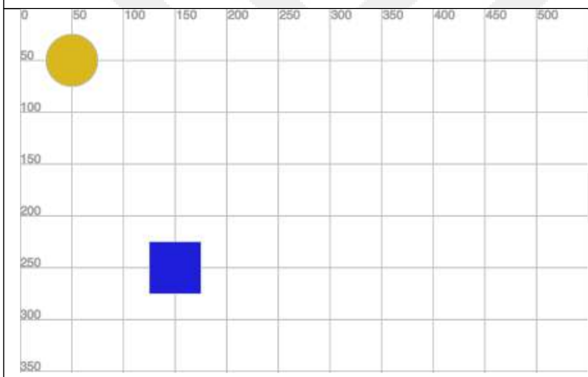
Style cards determine the attributes (color, position and orientation) of objects or animations. These cards include PICK COLOR, ROTATE SHAPE ORIENTATION, SET POSITION, and SET SIZE. The last category, animation cards allow LOOPING, ROTATION, and CREATION OF COLOR GRADIENTS. Using these commands, students can build new 3D models that show some simple animations. A simple program can be seen in Figure 4.5.

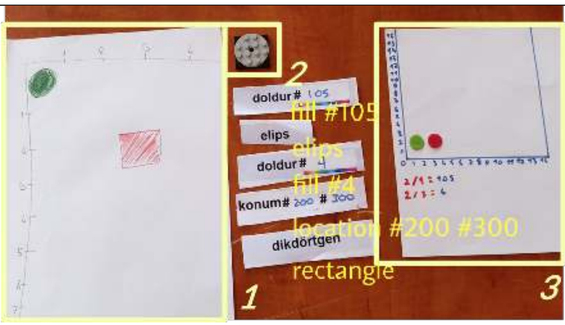
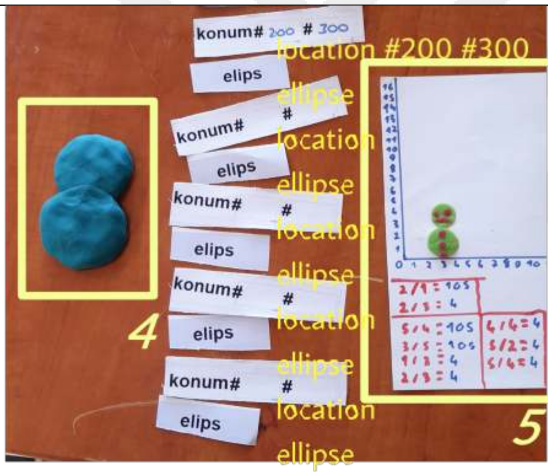
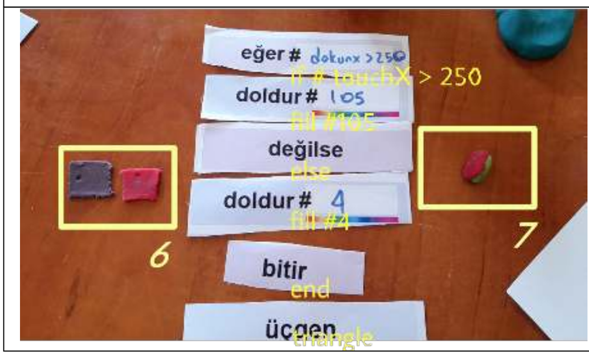
Scanning attribute cards such as PICK COLOR and SET SIZE results in a pop-up screen that globally defines an attribute. When the shape card is recognized, this particular 3D shape is rendered on the card with the determined attributes. For example, in Figure 4.5, the color card opens an RGB color picker pop-up interface. When students pick the color using this interface, this color attribute will be applied to shapes that come after this color card. In the end, the 3D model that applies the programmed attributes is rendered on the physical shape cards, similar to other marker-based interfaces.

4.3 Testing Students' Use of Self-Made Tangibles

In the exploratory design sessions, children created tangible representations of functions they were already familiar with in daily life, such as 'watering the garden.' Building tangible representations of these functions can be seen as an easier task when compared to more abstract programming blocks like conditional statements or loop structures. In this regard, my goal in the final study was to apply the design considerations to creating tangibles of paper-programming functions and testing their efficacy. First, I gave a programming task and asked children to complete it using Kart-ON programming commands. After creating each code, they ran it to see the output. Then, I asked them to create a tangible representation to save this code as a function.

Two children (Student A and B, both eleven years old) participated in this study. As they did not partake in the previous studies, they were not familiar with the "self-made" programming tangibles and had not heard about the design considerations before. I provided the same materials as preivous sessions (e.g., playdough, LEGO bricks...) and gave 3-5 minutes to create each tangible representation. During the session, I talked with the children about their intended design and helped them to keep track of the worksheet.

<table>
<tr>
<td>
</td>
<td>Students used the Kart-ON application, a paper programming environment, to create different algorithmic drawings in this study. This figure demonstrates how the tablet camera captures the paper programming commands When students run the code, they see the output on the left figure.</td>
</tr>
<tr>
<td>
</td>
<td>When students tap on the “Compile and Run” button on the bottom-left corner of the screen, they see the drawing above. The application displays the resulting drawing on a coordinate system to help students better understand setting up the locations of the shapes. This figure shows an amber ellipse and a blue rectangle. The ellipse is displayed in its default location, as the user only determined its color but not its location in the code. The rectangle applied the x and y coordinates of the location command, which are 125 and 225 in the code.</td>
</tr>
</table>

	In creating a tangible representation task, both participants tried to replicate the output of the code created earlier on the left in the coordinate system. Box#1 and #2 are created by Student A. The LEGO bricks were added when Student A noticed that he did not apply DC#3. Student B forgot the shape used in the earlier code, which was square, and created a circle with playdough instead (Box#3).
	Box#4 has two blue play dough ellipses made by Student A. Yet the student did not want to continue with the coordinate system graph. Student B used the same graph and created a green snowman in Box#5.
	Box#6 uses two play-dough rectangles to show the color-changing operation. Box#7 is a one-piece play dough that contains two adjacent triangles.

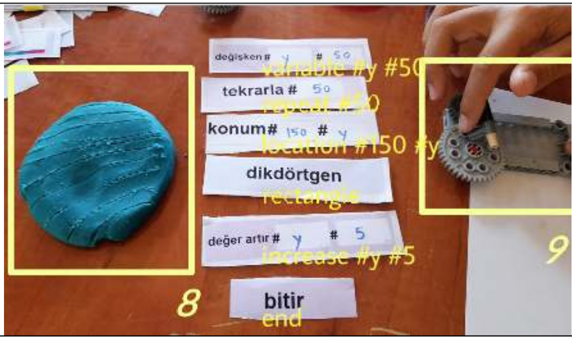
	Box#8 has 50 lines that indicate the loop count. Box#9 is a LEGO gear that completes many cycles and stops when the loop count is completed.

Table 4.1: **Using self-made tangibles together with Kart-ON**: Session flow and design outputs of the session where students applied the design considerations in their self-made tangible creations.

I summarize early insights of the design considerations (Section 3.5.2) in action to create tangible representations for paper programming codes. Although the limited participant number restricts the generalization of findings and early observations demonstrated that the playful nature of the tangible building phase made it difficult for students to follow the design considerations.

- **DC#1 (Giving hints about the action):** As Program #1 and #2 do not state any action and only produce static images, students naturally could not give hints about the action. Program #3 was an if-then-else structure requiring checking a touch input, but students did not hint about conditional structure in Box#6 and #7. Both students represented the repeating action for Program #4; Box #8 has rotation capability, and Box #9 has guided students to use materials that can increase lines that demonstrate loop count.

- **DC#2 (Choosing the right materials):** Eventhough I guided students in choosing the durable materials, they tended to use their favorite and more familiar materials, which were LEGO and play dough. Choosing the right materials for future use cases was the hardest design consideration to follow.

- **DC#3 (Adding details with simple shapes):** All tangibles use simple shapes, which helped students to complete the tangible creation task in the given time. I could run the paper codes and create a tangible representation for four different tasks in under eighty minutes.

- **DC#4 (Combining materials):** The first program uses a coordinate system graph and play-dough, which combines different materials and gives details to the users. But, in the second program, Student A did not add the coordinate system and continued with an abstract representation of a snowman. When I reminded him of design considerations, he stated that this version was enough to remember the code. Through the study, the number of combined materials decreased, and students tended to focus on one material in their design.

Overall, these design considerations helped us to manage the duration of the design session. But, students could not fully comprehend nor integrate the design considerations into their representations. I conjecture that they might need more practice and guidance on how to follow them through, which I will consider in the further studies. The challenge of maintaining the balance between playfulness and developing a functional self-made tangible will be addressed in future work.

4.4 Testing Usability, Engagement and Collaboration

I conducted the user studies in two steps. First, I held several individual and group studies to improve the usability in an iterative approach during the COVID-19 restrictions. Then, I held three in-classroom studies using three main research outputs (Kart-ON, AR mode and Pen mode) to see the products in action. During the studies, I carried out semi-structured interviews.

4.4.1 Participants and Research Settings

In the first study, I conducted iterative user studies with a total of fifteen students ($\mu_{\text{age}} = 10.93, \sigma_{\text{age}} = 0.88$ — 6 females, 9 males). In this step, I used Kart-ON to gather information about the usability of our card-based methodology. After each session, I updated the application based on students' feedback. So, in each study, students used a slightly changing version of Kart-ON. Due to COVID-19 restrictions, these studies were held in various settings like schools and parks.

In the second study, I visited a public school and randomly assigned each tool (i.e., Kart-ON, AR mode and Pen mode) to three classrooms. Some of the students had experience with programming, but most of them had no previous experience. Students who participated in this study were 6th graders at a public school in Turkey. A total of ninety-one students ($\mu_{\text{age}} = 12.13, \sigma_{\text{age}} = 0.39$ — 43 females, 48 males) participated to the user studies. Group 1 (Pen Mode) consisted of 30 students (43% girls), Group 2 (Drawing Mode) consisted of 31 students (48% girls), and Group 3 (AR Mode) consisted of 30 students (50% girls). Due to the conditions at the school, the participants could not be randomly assigned to the conditions. However, all these groups came from the same school to control the school and teacher differences. In each classroom, they had access to a limited number of tablets. Students who can bring their own or family's tablets used them with their group. A summary of the user study flow can be seen in Figure 4.1.

4.4.2 Procedure

Throughout the first set of iterative studies, students used different versions of the programming environment, but completed similar programming activities. First, I gave definitions of fundamental terms like programming and algorithms. Later, I handed them basic programming cards. Before handing a card to students, I asked their guesses on the function of the given card. Prior to programming in a smartphone, I mostly showed the step-by-step execution and possible output of the algorithm on the board, then used a smartphone to run the algorithm via paper cards. After introducing the setup, students worked on several programming activities and tested their programs using their tablets.

In both studies, I followed a similar programming activity. First, I gave definitions of fundamental terms like programming and algorithm. Then, I introduced some real-life examples in an algorithmic structure. Later, I handed them basic programming cards. For example, in Kart-ON case, these cards contain drawing commands like *fill*, *location*, *size*, and *ellipse*, whereas in Pen-mode, the cards are *forward*, *left* and *right*). Before handing a card to students, I asked their guesses on the function of the given card. Prior to programming in a smartphone, I mostly show the step-by-step execution and possible output of the algorithm on the board, then used a smartphone to run the algorithm via paper cards. After this introduction to the setup, students worked on several programming activities. I also integrated a worksheet activity while introducing HSL color definition, where students used the mobile color picker interface to predict the *H*, *S*, and *L* values of given color on the worksheet. This activity aimed to teach students the basics of sequential thinking and helped them to understand using commands to draw shapes on a screen. In all the sessions, the main researcher helped when students asked but otherwise remained as the observer. The sessions lasted around one hour and were documented as field notes.

In the classroom studies, I asked students to bring an Android tablet if possible. In each classroom, I had enough tablets to form groups of five, which allowed us to teach the basics of programming and prototype their ideas freely. Although I followed similar procedures in these studies, I tested different variants which resulted in different outputs: *Kart-ON's output* was a rectangle and ellipse with different colors. I improved the output by adding touch input as location information. *Pen Mode's output* was drawing a rectangle first, then drawing a star by changing the inputs of commands. *AR Mode's output* was drawing three spheres with a color-changing animation.

4.4.3 Measures

Collaboration was measured with six items on a 5-point Likert scale [Fu et al., 2009]. The social interaction subscale of the original scale was based on e-learning games; therefore, I adapted the items according to the purpose of our study (example item: "The app I use allows collaboration between me and my friends"). The items were translated to Turkish using the translation-back translation method. The internal reliability of the scale was .88.

User engagement was measured with nine items on a 5-point Likert scale [O'Brien and Toms, 2013]. The original scale items were related to user engagement after using an online search platform. I adapted the items according to the purpose of the study (example item: "I enjoyed using the application"). The items were translated to Turkish by using the translation-back translation method. I had to remove the reverse-coded items from the score calculation because the results showed that children did not understand those items. After looking at the responses to the other questions on the scale, it was expected that children should answer those questions with "definitely disagree" or "disagree." However, most of them responded with "I am not sure," which signals that they did not understand negatively worded questions. Exploratory factor analysis findings provided that those reverse items did not load to the same factor as the other items (.33 and .28 were the loadings for the first factor).

Technology interest was measured with five items on a 5-point Likert scale [Denner et al., 2014]. The attitude toward computing subscale from the original article was used, however, the questions were explicitly on computing. I adapted the questions to cover general technology interests (example item: "It's fun to learn what technology does and can do"). The items were translated to Turkish by our research team, and the translation-back translation method was used. The internal reliability of the scale was .87.

Appendix C shares the questionnaire that we applied in the classroom studies (in Turkish). In the analysis, we utilized one-way-ANOVA to compare the means of the groups. Each measure item (collaboration, user engagement and technology interest) is compared between the groups (Kart-ON, Pen Mode, AR Mode, and Control) to see if there is a significant difference between the means.

4.4.4 Notes on COVID-19 Restrictions.

Our main programming flow was designed to boost collaboration and active discussion in the classroom. However, new health regulations promoted keeping social distances between students. The current health concerns still suggest these interaction rules, which affect the usage of all TUIs in the classroom. In our group studies, I warned children about keeping their distance from each other in several circumstances, which also shows their intention to collaborate.

4.4.5 Data Collection and Analysis

I employed three main data collection methods: (1) I took observational notes of the field study, (2) logged the mobile app activity, and (3) conducted semi-structured interviews with the students. I could not take video recordings of the study as our national education department does not permit it. So, I resorted to taking field notes that revolved around the participants' participation, interaction, involvement, enjoyment, and enthusiasm. I particularly noted students' facial expressions, posture, gestures during the study, as they are commonly used factors to understand the behavioral and cognitive engagement in mobile learning applications and [Yun et al., 2020, Bond et al., 2020]. After the studies, I also conducted semi-structured interviews about students' regular tech usage, challenging spots using the app, and the parts they mainly enjoyed. In addition, all mobile app sessions are recorded using UXCam [UXCam, 2021] to measure the screen time and see the screen interaction.

I performed open coding and inductive reasoning in the qualitative data analysis. I reflected on field notes and analyzed written answers of semi-structured interviews to understand the effects of our application. I summarized our findings in two main branches: (1) Effect of using these environments on students' collaboration and engagement, (2) Benefits and challenges of providing a physical setup with paper cards and working with AI tools.

(a) (b)

Figure 4.6: **Setup images from different user studies**: (a) Programming commands from the field and at-home studies. (b) User study setups from individual and group meetings.

Baseline Analysis

One-way analysis of variance (ANOVA) results showed that the three study groups did not differ in their current technology usage (i.e., daily computer, tablet, and phone usage) and previous participation in coding education at baseline [F(2,87)=.492, p=.61, F(2,88)=1.448, p=.24, respectively]. The Chi-square test indicated that sex distribution did not differ between groups,

$$x^2(2) = .29, p = .86.$$

Collaboration

ANOVA results showed that there was a significant difference between groups on collaboration, F(2,84)=14.06, p¡.001. Post-hoc analyses indicated that Pen Mode group (M= 4.20, SD=.63) and AR Mode group (M=4.45, SD=.55) scored significantly higher on collaboration than Drawing Mode group (M=3.40, SD=1.08), p¡.001. Pen Mode group and AR Mode group did not differ from each other, p=.45.

User Engagement

ANOVA findings indicated a significant difference between study groups on user engagement, F(2,85)=5.50, p=.006. Post-hoc analyses indicated that Pen Mode group (M= 3.90, SD=.74) and AR Mode group (M=4.00, SD=.56) scored significantly higher on user engagement than Drawing Mode group (M=3.38, SD=.96), p=.03, p=.008, respectively. Pen Mode group and AR Mode group did not significantly differ from each other, p=.88.

Technology Interest

ANOVA results showed a significant difference between groups on technology interest, F(2,84)=5.26, p =.007. Post-hoc analyses indicated that Pen Mode group (M= 4.41, SD=.67) and AR Mode group (M=4.54, SD=.67) scored significantly higher on their technology interest than Drawing Mode group (M=3.88, SD=1.05), p=.04, p=.008, respectively. Pen Mode group and AR Mode group did not differ from each other, p=.83.

4.5 Results

4.5.1 Collaboration

In the analysis,the term *collaboration* is defined as a mutual effort to learn, which involves a constructive conversation, asking topic-related questions, guiding the conversation, and reflecting on ideas [Dillenbourg, 1999]. In the group studies, students actively talked about the given tasks, reflected on each others' ideas, and challenged the given task, which fulfills the collaboration goal. For example, in Pen Mode classroom study, after learning to draw a rectangle, one group continuously challenged each other, which resulted in drawing several different shapes. Unlike other versions, AR mode increased the conversations *between* groups; they competed to create new 3D outputs. The novelty of the modality led students to display their outputs to each other and discuss how they achieved them.

UXcam logs revealed that in a one-hour study, active time spent on coding and output screens varied between 5-10 minutes. The time spent indicates that I can utilize one smartphone/tablet among 6-12 groups of students in an ideal classroom environment. As one can imagine, the group dynamics and previous experience determine the quality of the session, and the required amount of tablet devices.

In an ideal environment, one smartphone can be enough to complete tasks in a classroom. Yet, providing a smartphone for each group and continuously testing the codes might be more suitable for another classroom.

Only two out of twenty-two groups had struggles in collaboration. In both cases, the tablet owners did not allow others to use their tablets, which resulted in negative experiences for other group members. To solve the issue, the lead researcher landed them his own tablet and helped them run the program. But, this situation showed us that, organizations that would like to use shared tablets in classroom should not depend on children's own tablet.

4.5.2 Engagement

Blending the movement and action in the physical space with digital programming is an ongoing challenge for TUIs [Fernaeus and Tholander, 2006]. The transition between these domains can interrupt the interaction and result in a less engaging environment. Yet, in the program building phase, I observed that this interruption occurred naturally as it feels like completing one task (algorithm building), and moving to another (compiling the program/debugging).

In addition, students' movements such as reaching the cards on the table, moving around the program blocks, and showing enthusiasm to participate in discussion demonstrate the direct effect of screen-free time. The observers note that students kept their excitement throughout the session as apparent in their demeanor (upright, gaze on the task) and continuous conversations [Yun et al., 2020]. Students discussed the programming concept while holding the programming card, switched it amongst themselves, and rearranged it on the table when stuck. Such physical interactions increased the involvement of even the most reluctant participants, who refrained from making comments but demonstrated their idea 'tangibly' in the conversations. Therefore, the engagement of the participants was kept alive with hands-on conversations.

Although three variants resulted in similar engaging experiences, using the AR version particularly excited students when they saw the program output 3D on a paper card. I heard most "*wow*" s when they created the output on 3D mode. Nevertheless, this can be a *novelty effect* and may not engage students in a long-term curriculum setup.

4.5.3 Providing a Physical Setup

I believe that the physical setup endorsed children's process-oriented nature, as their primary attention was not on the digital output. The user studies and UXcam results revealed that students spend most of their time in active discussions and negotiations on the final version of their code before trying these out in the mobile application. During our on-lecture conversations, students mentioned that sharing the smartphone came naturally, as they use smartphones to play games and watch videos together. Also, the students tended to divide the cards amongst themselves to find the right one and switch these like tokens. As it is a familiar material that children are already using to pass information and a natural medium for collaboration, they figured out easily how to use the card. The physicality of the setup also engaged reluctant students who refrain from making comments and prefer to demonstrate their ideas in a hands-on manner. They were always aware of their current tasks. The nature of card-based programming methodology requires keeping a neat workflow by scanning cards step-by-step. Our observations revealed that holding a single card and discussing the content step-by-step on the table helped students to focus.

The card-based programming led us to organically integrate worksheet activities in our workshop flow. Introducing the HSL definition of colors with small gamified activities increased their engagement. All students enjoyed the activity and wanted to continue it even after finding five different color definitions. Therefore, integrating interdisciplinary content with gamified activities is an opportunity in Kart-ON to help children interactively learn scientific concepts. Further, cards brought gamification ideas in the tangible domain. For example, I asked students to complete the missing cards to complete some tasks, which appeared as an exciting challenge. Teachers can use these kinds of activities and board-game-like activities, which can naturally be suitable for a card-based interface.

The main drawback of using paper cards appeared when the cards were deformed, and the mobile application could not recognize the card. I can solve this issue by printing them on a rigid material. This material choice can decrease the affordability, but it can significantly improve the user experience.

4.5.4 Working with AI tools

Powering the application with computer vision algorithms is both a challenge and an opportunity. For example, in some cases, when students encountered unrecognized illegible handwriting, they could not realize this, and the output differed from their expectations. I observed the need to improve handwritten text recognition accuracy on programming cards. Figure 4.6-a shows some examples of children's programming cards. Students used different pen types and varying colors to fill the input field. To increase the recognition accuracy, I are collecting a dataset that contains programming card images. On the other hand, introducing the inner workings of Kart-ON was an opportunity to build awareness of AI to children, which is becoming more and more important in today's world. In the studies, they could understand when and why the system did not detect the command or their input. I mention how character recognition data has regular lighting and style, so the model can only accurately predict this kind of input.

4.6 Design Considerations

Developing the tangible programming environment in distinct syntax, modalities, and user groups allowed us to observe the potentials and the limits of paper programming, along with the usage requirements and necessary changes that led us to compile design considerations for practitioners. The development and test experiences guided us to curate six design considerations for developing accessible paper programming environments. Researchers can combine these design considerations with other construction guidelines for children, such as Resnick and Silverman's [Resnick and Silverman, 2005], which provide ten principles to help children become more fluent, expressive, and comfortable using new technologies.

1. **Choose card designs based on in-class activity flows.** One of the potentials of paper programming cards is their physical variance. They can be in multiple shapes, textures, and colors. Yet, current implementations are limited since most applications use QR codes and predefined shapes to recognize these cards. I suggest diversifying the cards by carefully considering the activity at hand. For example, Kart-ON activities require combining many programming cards vertically on one screen. To support this physical arrangement, I designed the card dimensions accordingly and used color as a distinguishing factor. The number of possible vertically stacked programming cards depends on printing size and the arm length of students. The maximum number of stacked cards in our studies was eight. I promoted stacking the cards based on the functionality of the given command set.

   Alternatively, the AR cards were designed like a real-time prototyping tool that uses one programming card at a time, so the cards can be larger and more diverse. Even though I only used square cards, other shapes can be considered to make children recognize and distinguish cards more easily. Current object and text recognition models and fine-tuning these models with the transfer learning approach [Tan et al., 2018] can yield accurate results when custom design cards are employed.

2. **Find technically relevant input methodology.** The main challenge of paper programming is determining the proper input methodology. Handwriting recognition, introducing new tangibles with transfer learning approaches, placing new NFC stickers to new tangible inputs can be all appropriate for different purposes. Kart-ON uses handwriting recognition for inputs as the inputs have limited character amount, which results in acceptable error counts. AR version utilizes a smartphone screen to enter new inputs as it actively involves mobile devices.

3. **Define abstractions in the same modality.** In the physical computing domain, defining abstractions generally occurs via using mobile devices or computers. This situation increases the dependence on electronic devices and reduces affordability. Designing self-made tangibles for function and variable definitions decreased the total screen time, supporting our goal of achieving a more efficient model for sharing smartphones. However, I also observed that children struggled to create new tangibles to represent abstract definitions. I still need to explore students' self-made tangible design process and develop a more detailed guideline to ease the process.

4. **Choosing Black-boxes Carefully:** As Resnick and Silvermann describe, the abstraction-level of the kit defines the ideas that users can explore and remain hidden from view [Resnick and Silverman, 2005]. One of the main discussions during the development of our programming environment was choosing black boxes in high-level operations. For example, in the development of Kart-ON, our goal was to keep the drawing operations semantically similar to current creative coding frameworks from an educational perspective. In this way, when students encounter popular frameworks like Processing, they could become much more familiar. But, in the user studies, I observed that using four inputs in one command has drawbacks. So, I decided to add new functionalities in Processing-style command blocks.

5. **Consider the pedagogically appropriate number of student inputs.** Determining the right number of inputs for children's comprehension requires iterative experiments. Our observations revealed that requiring more inputs in command has two drawbacks: (1) Students struggle to keep all the required inputs in mind, (2) Educators spend a substantial amount of time before using only one command. In our experiments, I first printed the Kart-ON cards in the same style with Processing commands. For example, similar to Processing, our *rectangle* command was taking x, y, w, and h inputs. However, students had difficulty remembering all these different inputs in one command. Then, I

introduced two new commands to ease the shape commands: *size* and *location*. Each command takes two inputs, and students could easily understand the commands' roles in the code.

6. **Support teachers' adaptation to the system.** Having paper cards puts an extra amount of work on teachers' shoulders. Before studying with children, I regularly met with teachers in the development process, and they were anxious about the work required to implement this programming approach in their classrooms. They need to print cards, actively observe the groups, walk around student groups while scanning the cards and ensure the computer vision model yields the correct output. Before the development stage, listening to them led us to design various elements in our programming flow like "grouping behavior," "structured worksheets," and "multi-screen interfaces."

7. **Determine the physical workspace setup** The benefit of using tangibles in programming is keeping children physically active and enhancing collaboration. If the physical setup of the classroom cannot allow that, the tool cannot achieve these benefits. In our case, having a tablet for 4-5 students result in good collaboration, but asking students to bring their tablets is not the ideal scenario.

4.7 Implications

Kart-ON has the potential to increase collaboration and engagement while reducing screen time which is directly impacting the affordability of the learning setup. Most importantly, the user studies demonstrated that students enjoyed using our paper programming environments even if they spent limited time using a tablet. Students individually asked for further details about downloading and running the application at the end of the studies, which showed their motivation to use our app further.

The UX analysis results indicated that the Pen Mode group had a significant advantage over the Drawing Mode group in all measurements. Although the Pen Mode and Drawing mode use the same modality in input and output formation, Pen

Mode has a relatively simpler command set and uses an intuitive way of drawing. These results suggest that starting with a simpler set increases usability and motivation, which led us to the decision to use the Pen Mode activities in the early weeks of the curriculum. The Drawing Mode activities were used in the later weeks of the curriculum, when students had more experience with the Pen Mode activities.

Chapter 5

PROGRAMMING ACTIVITIES FOR A SEMESTER-LONG CURRICULUM DEVELOPMENT

Notes to Practitioners:

- Table 5.2 lists the main activities of the curriculum with the learning goals and open-source links. The activities are designed in line with the Turkish National Curricula.
- The long-term curriculum studies are conducted in collaboration with Koc University Psychology Department - Independent Evaluation Laboratory. The evaluation process is not part of the thesis.
- The curriculum activities and helper videos are available at our project website `https://karton.ku.edu.tr/karton`.

I developed a 12-week of curricular content following the main flow of the Turkish national curriculum. Programming education is part of the ICT (Information and Communication Technology) curricula of 5th and 6th graders in Turkey. Before the curriculum development process, I organized an ideation workshop, where I met with teachers, curriculum-makers, NGOs, and museum educators to discuss their needs and list possible collaborations. This chapter explains and discusses the ideation workshop process and its implications on curriculum design. The curriculum is designed to become an illustrative example to support tablet-sharing in the classroom and ease the teacher's classroom flow while using paper programming cards. All curricular activities are open-access and open-source, which eases the adoption of the curriculum by teachers and other education stakeholders.

5.1 *Ideation Workshop: Meeting with Education Stakeholders*

In parallel to the UX studies, I organized a workshop where I met with five stakeholder representatives in an ideation workshop setting. These stakeholders were two program coordinators from different NGOs, a private school ICT teacher, a national curriculum committee member, an educational content developer, and a researcher from a university. In the workshop, I discussed controlling the in-class flow efficiently, supporting different infrastructures, managing teachers' preparation time, and sharing best practices among stakeholders.

I conducted the ideation workshop meeting in three parts: (1) Introducing the project's goal and discussed the positives and negatives of ICT education. (2) Choosing one activity from the current middle school ICT book as a case study and listed the best practices and points that needed improvements. (3) Creating a new activity that reflects the analysis at the end of the workshop.

Limitations of the ideation workshop: I tried to engage many stakeholders from the education setting, but only five of them could attend the workshop day. In addition, although the focus was on developing a new inclusive curriculum for different socio-economic needs, teachers and other stakeholders mentioned many concerns about social media, national policies, and other subjects.

Workshop Analysis

I recorded the video of this workshop with their consent to further analyze the transcript. After the workshop, I thematically grouped the video transcripts and the notes. I formed three main categories: (1) Smoothing the change process, (2) Increasing school management's participation, and (3) Reducing the in-class workflow effort.

(1) Smoothing the change process. In the meeting, one of the main concerns of teachers was not feeling confident when they start teaching via new technology in classrooms. So, designing a new technology requires comprehensible tutorials for teachers from all disciplines if I would like to aim for the teacher's well-being in schools. Additionally, ICT is a continuously developing area, which brings exploring new content daily if they want to keep up to date. To smooth the change process in the curriculum, I included a beginner-guide to computer vision, examples for creating tangibles, and guides for transitioning to Javascript creative coding libraries.

(2) Increasing school management's participation. Teachers repeatedly mentioned that when students participate in competitions with an interdisciplinary approach like "solving a smart city's problem," they engage further in the activity. These activities offer unique opportunities where students can design a product so that they can do the marketing and on-stage presentations. In addition, these competitions are increasing the involvement of the school management in ICT activities. Participating in these activities increases schools' publicity, motivating them to invest more in ICT infrastructure. Although developing this kind of competition is the long-term goal, creative coding is one of the unique fields to offer challenges affordably. Students can create rich outputs only using a few inexpensive materials. So, I included a challenge at the end of each curricular activity and motivated teachers to share them with other teachers.

(3) Reducing the in-class workflow effort. Another direct impact of the ideation workshop on the development process was creating new helper tools for the development environment. All stakeholders found the smartboard integration into the classroom a great example of technology integration. Most teachers use smartboards daily; they show videos, examples, and book content via smartboards. So, I immediately developed helper tools specifically designed for smartboard use. For example, while using Kart-ON, students can create algorithms via paper cards, the teacher can show the outcome via smartboard, then students can scan these cards using a tablet, knowing which output they should see at the end of the scanning process.

5.1.1 Limitations

The ideation workshop depicted the concerns of a few education stakeholders in a local context and the illustrative curriculum is designed to be integrated into the

Turkish national curriculum. Although, I tried to solve local concerns throughout the design process, I also applied globally applicable frameworks, such as PRIMM to increase the adaptability of the curriculum. As the curriculum puts computational thinking and programming abilities into the core of the learning goals, I believe that researchers and policymakers can adapt their own paper programming activities easily. I shared all the materials open-source to accelerate the adaptation of programming education in disadvantaged areas.

5.2 Developing the Kart-ON Curriculum

The twelve-week illustrative curriculum can be easily adapted for any paper programming environment. In the curriculum design, I created activities to demonstrate the impact of computing with real-life examples and practice it with hands on activities. Tissenbaum et al. argue that the programming activity design should start with the end goals by considering various endpoints that are personally relatable to students [Tissenbaum et al., 2021]. In practice, teachers and policy-makers often use combinations of various knowledge sources to inform their decisions as they plan a new curriculum [Pareja Roblin et al., 2014]. Similarly, I informed the decisions with the design considerations and the PRIMM framework [Sentance et al., 2019]. PRIMM presents a sociocultural perspective to teach specifically programming. The approach draws on existing research and presents a lesson structure with five elements: Predict, Run, Investigate, Modify and Make. Following these design considerations and national curriculum standards [MEB, 2018], I developed a twelve-week curriculum for middle school programming education that focuses on using Kart-ON and shared mobile devices in the classroom. Since the curriculum is a variation of an already existing curriculum, I used PRIMM while designing new activities.

A curricular activity that includes programming via Kart-ON can be summarized as follows:

1. **Predict**: Groups of students grab paper programming commands and try

to order them to solve the given problem. Working in groups encourages students to discuss commands' functionalities and the program's behavior. This prediction step can also involve different unplugged activities, such as drawing the expected outcome on a coordinate system paper.

2. **Run**: When a group thinks the program is ready, they scan the cards and test if the output results in the expected outcome.

3. **Investigate**: In this step, students debug their program. If everything is right, the teacher can also ask how individual snippets of codes affect the program.

4. **Modify**: Students edit the program. This edit can include changing the inputs or commands.

5. **Make**: Students use their knowledge to solve a similar but new problem.

Considering the UX Results, the first two weeks of the curriculum consist of unplugged activities to focus on the basics of problem-solving methods. Then, for the next three weeks, students use Pen Mode, which focuses on sequential algorithm building and loop structure. During the Pen mode, they use Kart-ON's computer-vision power to scan physical programming cards. After students grasp the fundamentals of programming and have some experience with computer vision tools, I introduced the basics of computer vision and object recognition in a game-like environment for two weeks. Considering that using a computer-vision-powered programming environment is one of the new technologies teachers might find frustrating, I created activities that help teachers explain how computer vision works and how the programming environment leverages this new technology (Table 5.2). In the rest of the curriculum, they use Drawing mode. Additionally, I created a guideline showing how to translate their knowledge from Kart-ON to p5.js creative coding on the web.

Considering the study results of AR output modality on students' engagement, I included worksheet activities that promote real-time outputs in a collaborative

	Predict	Run	Investigate	Modify&Make
Pen Mode - Repeat	Use only repeat, forward, left, and end commands one time to draw a square.	Test the program using tablets.	Debug and test with different inputs. (e.g., forward #100 vs forward #200)	Draw a hexagon using the same commands.
Drawing Mode - Color	Use touch input commands in the “if” command to control fingertip position	Test the program using tablets.	Debug and test the touch position and operators. (e.g., touchx >350 vs touchx <250)	Create another if in the else scope to change between three colors.

Table 5.1: **Kart-ON curriculum**: Two example activities with their PRIMM-based steps.

manner. For example, in the color definition activity, students first try to find the numerical definitions of the printed colors using a color picker. Then, using the markers printed on the worksheet, students can see this color on a shape they picked. I aimed to establish the feeling of challenge and create tangible outputs whenever it is possible. In this activity, trying to find the right numerical definition of the printed color presents a challenge like a puzzle. Then, using markers depicts this numerical definition of a shape simultaneously.

Goals	Name and Link	Summary
G1.1, G1.2, G1.3	Unplugged Activities (Link)	Students solve two problems (Hanoi Towers and Wolf, Sheep, Grass) problems in an unplugged style.
G1.1, G1.2, G1.3	Pen Mode - Simple Drawings (Link)	Draw square and triangle using Pen Mode.
G1.1, G1.2, G1.3, G2.9	Pen Mode - Repeat Structure (Link)	Draw square and triangle using the repeat structure in Pen Mode.
G1.1, G1.2, G1.3, G2.9	Pen Mode - Advanced (Link)	Use pen commands in creative drawings.
G1.7, G1.8, G1.9, G1.11	Color Definitions (Link)	Representing colors.
G1.1, G1.2, G1.4, G1.5, G1.14, G1.15	Introduction to AI (Link)	Uses the Flappy Bird game development interface with Teachable Machine.
G1.1, G1.2, G1.4, G1.5	Drawing Mode - Shapes (Link)	Fundamental shapes to draw composite shapes.
G1.1, G1.2, G1.4, G1.5, G2.6, G2.7	Drawing Mode - Inputs and Conditionals (Link)	Use touch input in conditional structures.
G1.1, G1.2, G1.4, G1.5, G2.6, G2.7	Drawing Mode - Conditionals (Link)	Combine different boolean statements in conditional structures.

G1.1, G1.2, G1.4, G1.5, G2.6	Drawing Mode - Functions (Link)	Use function and call commands to store drawing a car.
G1.1, G1.2, G1.4, G1.5, G2.6, G2.8, G2.9	Drawing Mode - Functions and Loop (Link)	Use the car function under repeat structure.

Table 5.2: **The twelve-weeks curriculum summary**: The learning goals are adapted from Turkish national curricula (and listed detailly on Appendix A) [MEB, 2018]. The activity materials are shared open-source on Google Docs, and teachers can customized by their needs easily.

5.3 Implications

Section 3, 4, and 5 mainly focused on developing the programming environment that can support building an engaging and affordable programming education environment in classrooms. Teachers can complete the semester-long curriculum with a few smartphones or tablet computers and follow the worksheets with our helper tools to quickly adopt the system. Throughout the test process, I observed that the students were engaged in the activities and enjoyed the learning process. But, teachers stated that they were struggling to observe all students while maintaining the group activities. To understand and solve this concern, I aimed to create a helper system that can report the students' engagement in the classroom. Following chapters will describe the development of this system that can automatically detect the students' engagement in the classroom.

Chapter 6

COLLECTING THE GROUP ACTIVITY VIDEO DATASET FOR MULTIMODAL CLASSROOM ENGAGEMENT ANALYSIS

Notes to Practitioners:

- I shared the dataset, preprocessing steps and baseline analysis code, open-source at **Github Repository**.
- The dataset contains 26540 frames with self-engagement evaluation scores for each frame. I utilized recent state-of-the-art deep learning models to extract features from this audiovisual data. I also released the resulted OpenFace, and OpenPose vectors for the use of other researchers to decrease the carbon footprint in the replication process (see Table 6.2)

During field studies, educators and practitioners expressed concerns about their ability to observe the engagement levels of students during group activities. While Kart-ON and classroom tools have enabled programming activities to be completed in a cost-effective manner, the increased emphasis on group activities has led to an increase in workload for teachers managing groups of students. To assess engagement levels, experienced teachers combine multiple sources of information, such as affective, behavioral, and cognitive states of their students. Recent deep-learning techniques can also be used by researchers to reveal engagement patterns, which can help teachers in overcrowded classrooms. While some studies have analyzed engagement using only facial features, a multimodal approach to engagement assessment has yet to be deeply explored [Dewan et al., 2019]. Nevertheless, audiovisual recordings are a valuable resource for extracting information such as facial expressions, body language, gaze direction, conversational tone, and prosodic features. Expert educators can seamlessly use this information to inform their teaching, but novice teachers may require practice to effectively assess engagement levels. Therefore, automated recognition of engagement levels can be a helpful tool for teacher education or as an assistant in overcrowded classrooms in disadvantaged areas.

Yet, bringing a camera into the classroom and facilitating automation to augment teachers' abilities requires careful consideration [Saini and Goel, 2019]. Following a human-centered approach is vital to determine the challenges before a wide adaptation. Figure 6.1 summarizes the overall flow that consists of five main stages: Data collection, Exploratory analysis and feature engineering, Model development and fine-tuning, Ethics considerations, and Interactive application for end-result communication. Each part of the development process consists of unique challenges from machine learning and human-centered computing perspectives. In this chapter, I will describe the data collection and feature engineering stages of the end-to-end ML pipeline (Figure 6.1-1, 2 and 3). The next chapter, Chapter 7, describes the deep learning model development and Chapter 8 explains the interactive dashboard development for a student centric interpretation of model results.

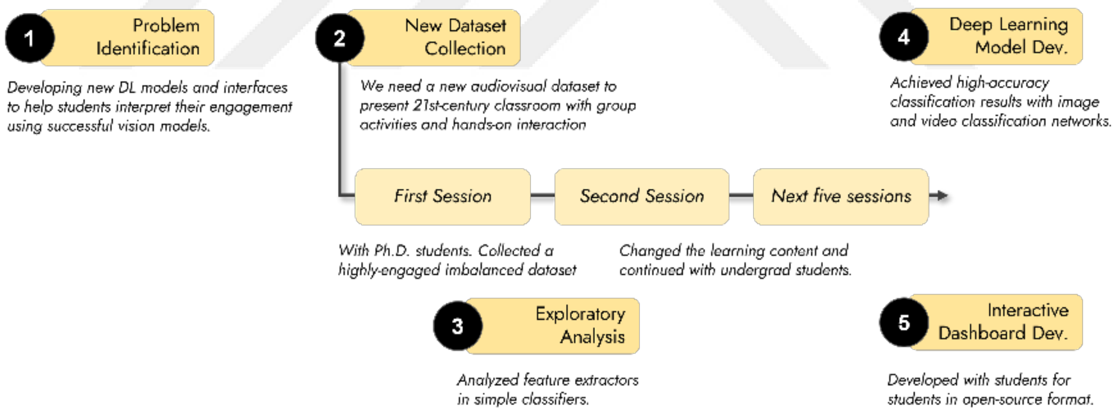


Figure 6.1: **The overall flow of the data collection and model development process**: The flow includes the data collection, exploratory analysis and feature engineering, model development and fine-tuning, ethics considerations, and interactive application for end-result communication.

To my knowledge, this end-to-end, video-to-interpretable feature extraction pipeline presents the most comprehensive open-source learning analytics data pipeline that utilizes the most recent state-of-the-art deep learning techniques. The research outcomes can accelerate the human-centered design of ML-based learning analytics.

6.1 Data Collection

Our dataset consists of group activities of undergraduate and graduate degree students, where they learn new creative coding tools via varying types of learning activities. The activity design reflects an active classroom setup that follows the educational approaches of a 21st century classroom. UNESCO defines this 21st-century learning environment as a classroom where students become active and engaged learners, thinkers, and creators, where they learn new things, experience hands-on interaction, and reflect on their ideas via constructive discussion [UNESCO, 2015]. Thus, in the dataset collection process, I aimed to represent these active learning activities in a group activity setup.

In the data collection process, I followed an iterative process similar to the my other research and development processes, which means, the activity design has slightly changes after the first two data collection sessions. But, to be able to utilize all the data in the model training stage, the activities followed a common theme, where all groups completed creative coding activities watching Youtube tutorials. Then, students completed some tangible, hands-on activities related to online learning content. Throughout the activities, they also answered some questions that automatically popped up on their tablets to follow their cognitive engagement. They also explored how creative coding applications can be used in research by experiencing some real-life examples.

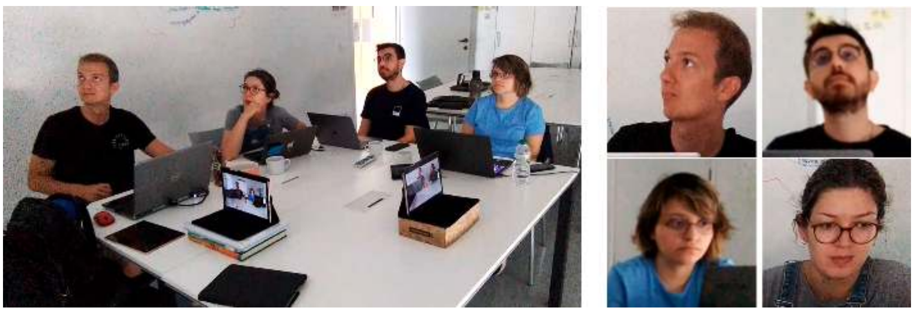


Figure 6.2: **Video recording formats of the dataset**: (a) Group-facing camera records the overall group activity to inform the model about the movement and shared attention, (b) Participant-facing cameras record faces directly to catch action units and emotions.

6.1.1 Overview of Final Dataset

Before explaining the participant selection, annotation, exploratory analysis and feature engineering steps, I presented the overview of the dataset in this section. The dataset contains videos from seven sessions. Total of thirty four students participated in the data collection studies (Section 6.1.2). I followed an iterative process and updated the activity design after conducting exploratory data analysis for the first two studies. (Section 6.2). Overall, the released dataset contains 26540 frames (FPS = 1, 7.5 hours) of audiovisual recordings with self-evaluation scores for each frame. In the self-evaluation process, they marked their engagement level between -100 to 100 using a slider that collects the annotation for each frame (Section 6.1.3). For each frame, individual faces are centered and cropped to 320x320 images. For the video action recognition model, ten-second-long video clips with two-second paddings are sliced. I automatically extracted dialog transcripts using Whisper and manually checked these transcripts, thanks to volunteers. Table 6.2 shows the explanations and open-source links for available features of this dataset.

In the end, the frame counts for each engagement level in a five-level classification task are:

- **Highly engaged:** 5914 (22% of all data)
- **Engaged:** 5489 (20% of all data
- **Moderate:** 7888 (30% of all data)
- **Disengaged:** 4271 (16% of all data)
- **Highly Disengaged:** 3084 (12% of all data)

The dataset is publicly available with some constraints. Four out of thirty-four students did not allow sharing the video recordings. Thus, I cannot publicly share the video recordings of three sessions. But, the models are trained using the full dataset. The video recordings, frames, OpenFace and OpenPose features are ready-to-download via script. Researchers holding a valid and up-to-date *Ethics Committee Approval* can also request the private links.

6.1.2 Participant Selection

In the participant selection part, I set three requirements to collect life-like data from an active learning environment:

1. *Knowing each other before:* I set this requirement to create a similar workflow to a classroom environment. In most classroom learning environments, groups of students already know each other and have things to share other than classroom-related topics.

2. *Having an introductory-level knowledge on topic:* If a participant already knows the topic, the learning process would be boring and disengaging for this participant. And, if the topic is very new to them, everything throughout learning process can be engaging and interesting.

3. *Having a teaching experience:* Part of the data analysis includes self-evaluation, which can be a challenging task if the participants have no experience of learning assessment. Having previous experience in teaching increased the ability to self-evaluate.

Additionally, collecting data from adult participants allowed us to publish the data publicly. Publishing learning environment data from the K-12 setting requires extra attention, and I could only publish it to the researchers with valid ethical committee permission if parents and children would allow the distribution of their data. These advantages make it easier for researchers to conduct studies on a larger scale, which can lead to more significant findings. Obtaining ethical approval from Institutional Review Boards (IRBs) is easier since university students are considered adults and are better able to understand the risks and benefits of participating in research studies. Furthermore, university students are often better able to evaluate their own engagement levels and provide more detailed feedback on the effectiveness of educational technologies [Andrade, 2019]. This is because they have more experience with educational technologies and are more likely to be reflective about their learning experiences. As a result, they may be more accurate and reliable sources of data for researchers to use. As all the participants are older than twenty-two years old, I could publish the dataset in open-source format with their permissions. The collection complies with GDPR regulations, which means the participant holds every right to their own data, including removing their identity-exposed parts.

6.1.3 Engagement Level Annotation

Each participant evaluated their self-engagement using a scale between -100 to 100 using the CARMA annotation software [Girard, 2014]. I prepared an *Engagement Analysis Checklist* to help participants evaluate themselves more accurately. Our analysis checklist is a subset that includes observable inventory items from Wang et al.'s *Classroom Engagement Inventory* [Wang et al., 2014]. Table 6.1 lists the selected checklist items and their engagement category based on Fredricks et al.'s definition. I included fourteen items from this inventory that can be physically observed by an independent evaluator. I reminded participants to regularly control the checklist to remind themselves of the definition of engagement and how classroom engagement is assessed. I also asked participants to complete a survey that asked about their research interests and their motivation to learn the topic.

6.1.4 Dataset Format and Pre-Processing

Raw videos. I recorded the videos using three Samsung A8 (2022 Model) tablets in 1280x720 resolution and 30 FPS. One tablet was recording the group from an upper corner. The other two tablets were directly looking at the students that sat side by side. The organization of the cameras can be seen in Figure 6.2. Each video is synchronized by using a clap sound at the beginning. After the clapping moment, I sliced the video based on their respective active materials/topics/discussions. I sliced the videos around 5-10 minutes to ease the annotation process. Slicing allowed annotators to manage their time efficiently and us to start processing the annotated videos more quickly.

Face Areas. Then, I cropped and resized the facial areas in order to obtain single-face videos from these two cameras. I used MediaPipe[1]'s face detection module to detect the center of the face area. Then, I used FFmpeg[2] to crop it by 320px to 320px, which is a commonly used resolution for facial action unit recognition models. In the audio extrac-

[1]https://google.github.io/mediapipe/

[2]https://ffmpeg.org/

Eng. Category	Item
Affective	Feeling interested
Affective	Feeling proud
Affective	Feeling excite
Affective	Feeling happy
Affective	Feeling amused
Behavioral	Listening very carefully
Behavioral	Paying attention to the things supposed to remember
Behavioral	Completing the assignment
Behavioral	Getting really involved in class activities
Behavioral	Actively participating in class discussions
Behavioral	Working with other students and learning from each other
Cognitive	Trying to figure out where the things went wrong
Cognitive	Searching for information from different places and thinking about how to put it together
Cognitive	Judging the quality of the ideas or work during class activities

Table 6.1: **Engagement checklist items and their corresponding categories**: I handed out this checklist to the participants while the self-evaluation process to help them memorize the engagement analysis process.

tion, I applied a noise filter to reduce the background noise during the *wav-conversion* process. In the transcription process, I first run OpenAI's Whisper model [Radford et al., 2022] to speed up the transcription. Then, I corrected the transcript text. Figure 6.3 summarizes these processing steps.

Engagement Level Labels. Each participant evaluated their self-engagement using a scale between -100 to 100. For the engagement-level model development, I created a five-level engagement system (highly disengaged (HD), disengaged (D), moderate (M), engaged (E), and highly engaged (HE)). I mapped the values between $[-100, -60)$ as highly disengaged, $[-60, -20)$ as disengaged, $[-20, 20)$ as moderate, $[20, 60)$ as engaged, and $[-60, 100]$ as highly engaged. I used these discrete levels in classification tasks.

I also created a binary engagement classification system by mapping the values between $[-100, 0]$ as disengaged and $(0, 100]$ as engaged. I used binary levels for only exploratory analysis (see more details in Section 6.2).

For the group level classification tasks, I determined the overall group engagement levels using minimum pooling. For example, if the labels are HE, E and D, the group engagement is D, as it is a disengaging moment for one group members, but other members were not affected from this situation.

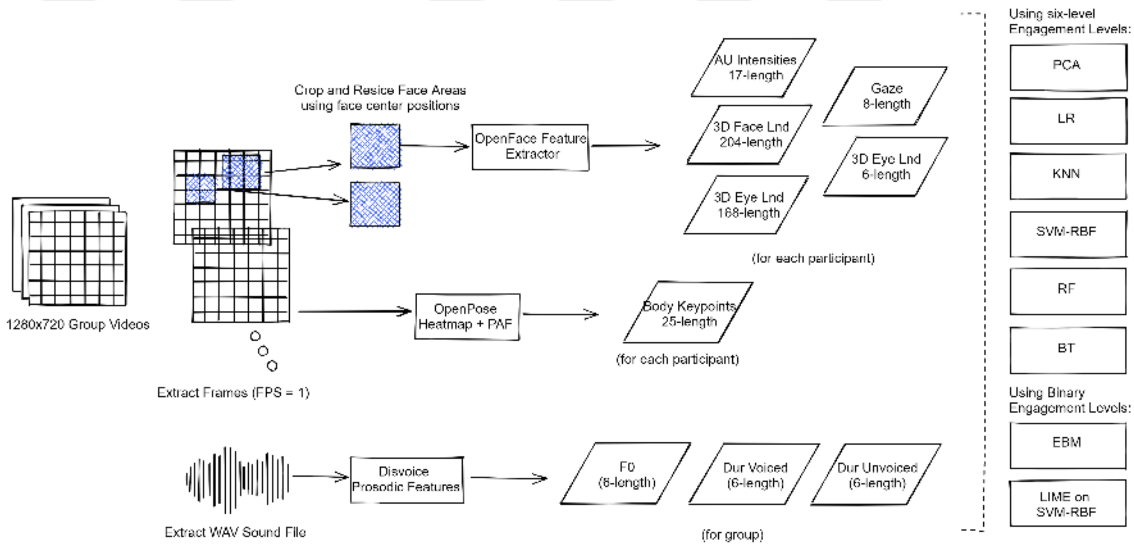


Figure 6.3: **Dataset collection and feature engineering flow**: The illustration of the data pre-processing and exploratory analysis pipeline with feature extraction steps.

Feature	Explanation	Link
Video Recordings	From seven different learning sessions, total 7.5 hours.	Link
Image Frames	Image frames of video recordings (FPS = 1).	Link

Self-Evaluation Scores	The scores are between -100 and 100. Each participant scored their classroom engagement following the *Classroom Engagement Inventory*.	Link
Engagement Levels	Self-evaluation scores are automatically mapped to five-level classroom engagements: Highly Disenaged, Disengaged, Moderate, Engaged, Highly Engaged.	Link
Video Slices based on Engagement Levels	Slices of videos for the continuous engagement levels.	Link
Face Frames	320x320 image frames of each participant. Centered and cropped.	Link
OpenFace Features	Combined OpenFace feature CSV for each video.	Link
Body Keypoints	Combined body keypoint positions for all frames.	Link
OpenPose Features	OpenPose Heatmap for different scales and keypoint JSON for each frame.	Link
Audio	WAV file - I applied noise reduction on the original audio.	Link
Transcript	Automatically transcripted to OpenAI's Whisper, then manually checked by humans.	Link

Table 6.2: **Media and feature set for each session of the dataset**: This table contains links for an example session which I used in the exploratory analysis (Session 2). All dataset and features can be downloaded via Download Script

6.1.5 Limitations

Self-Evaluation of Engagement: Self-evaluating the engagement level is challenging, as defining engagement in all aspects is still challenging for most researchers. Before the self-evaluation process, all participants completed a fifteen-minute training session that introduced the definition and dimensions of classroom engagement. Unlike the existing research, the system particularly focuses on capturing the multimodal aspect of classroom engagement. The labels of the dataset are ambiguous, and this ambiguous label represents a multi-dimensional complex term, 'engagement.' I collected data from group activities where students follow Youtube tutorials, hold group discussions, and complete hands-on, tangible activities. In this sense, the dataset is the only dataset that is collected in a controlled environment that represents the dynamic nature of learning.

Representation Bias: The collected data is from a small group, representing a very limited analysis of a small community. Considering these limitations, this data is limited for model training purposes but presents a unique dataset that is useful for exploratory analysis and testing.

6.2 Exploratory Analysis of First Batch

After the first two data collection studies, I conducted an early analysis to update the data collection process and resolve the issues as soon as possible. The main issue of the first batch was the imbalance of the engagement levels. The 70% of the frames were labelled as highly engaged in self evaluation, which presented a highly imbalanced dataaset. Following the exploratory analysis of the first two sessions, I updated the learning setup, participant selection criteria and activity flow to reduce this imbalance. In this learning setup, computer science students learning a new creative coding software. They are actively using their programming skills, but they have no previous experience in creative coding. So, the topic can be boring as they already know some programming, but at the same time engaging, as they learn a new concept in programming. This approach worked and the new activity flow achieved more balanced data that shows a normal-distribution like engagement-level category distribution.

The following section presents an example analysis for the Session #1. Three participants joined to this study, and their results are anonymized using their initials (B, C and Y).

6.2.1 Research Questions

In this exploratory analysis, I was trying to understand the nature of the data and how to achieve the best accuracy with the minimum data. Collecting a minimum amount of data is especially important as the future models can be trained using children's data. Throughout my analysis, I was trying to answer the following questions:

- Can I predict engagement level using AU Intensities or other observable face features?
- Can I predict engagement levels by using the joint positions of the participants?
- Can I predict engagement level by combining different observable features?

6.2.2 Methodology

To obtain the required features to answer these research questions, I used OpenFace to extract facial action units and OpenPose to obtain the joint positions of students. OpenFace's Face Feature Extractor yields 1562 features per vector. But, I analyzed a subset of five continuous feature sets: Face Action Unit (90 features), 3D Eye Landmarks (165 features), 3D Gaze Positions (9 features), 3D Head Positions (6 features), and 3D Face Landmarks (201 features) and one discrete AU Classes Feature Set (90 features). And, I extracted joint position information using OpenPose, which results in Body Pose KeyPoints (75 features) and HeatMaps (368480 features = image size).

I extracted one OpenFace feature vector and one OpenPose feature vector for each frame (FPS = 1). Before conducting further analysis using a feature set, I cleaned the data by thresholding the confidence score of face features. I removed all feature vectors that have less than a 0.5 confidence score. I also removed the vector if the row contains only zeros. After cleaning the data, I obtained 5351 frames out of 6969 frames. Then, I followed a three-step analysis strategy:

1. For each feature set, I applied PCA (Principal Component Analysis) to see the possible linearly separable features that can increase the overall engagement level classification. Principal Component Analysis (PCA) aims to reduce the dimensionality of a set of data consisting of variables correlated with each other while keeping

the variation in the dataset [Jolliffe and Cadima, 2016]. The variation in the principal components decreases while moving from the first component to the last one, and I can measure the variation explained with the PCA's dimensionality reduction by cumulative explained variation. As Open Face's Action Units result in discrete observations, I utilized MCA (Multi Correspondance Analysis) instead of PCA.

2. I trained five classifiers, Logistic Regression (LR), k-Nearest Neighbours (kNN), Radial-Basis-Function Kernel Support Vector Machine (RBF-SVM), Random Forest (RF), and Boosted Trees (BT) using individual and combined feature sets to analyze the impact of features and model characteristics on engagement level prediction. I trained these classifiers for five-level engagement classification and binary engagement classification.

3. Finally, I conducted explainability experiments to find the impact of individual features and test the combination of these important features on the same classifiers. I utilized InterpretML [Nori et al., 2019], an open-source Python library, to explain the behavior of existing systems using LIME. A drawback of the current implementation of InterpretML is its limited availability for multi-class classification. So, I only run the models that are trained for binary classification.

Our open-source repository contains codes for conducting these exploratory analysis step by step. Researchers can easily conduct analysis on their custom dataset or customize the code to conduct different experiments on the dataset.

6.2.3 Dimensionality Reduction

Figure 6.4 demonstrates an example PCA analysis for Participant C's OpenFace features. As the figures clearly show, the feature space is not linearly separable, but Head Pose, 3D Eye, and Face Landmarks formed some clusters compared to other features. I also presented the AU co-occurrence of Participant C in this figure. The highest co-occurrence value for all participants is 0.3, which shows no clear indication of correlation between action units for this experiment.

In addition to PCA, t-distributed stochastic neighbor embedding (t-SNE) and Uniform Manifold Approximation and Projection (UMAP) are non-linear dimensionality reduction techniques that focus on preserving the local structure of the data [Van der Maaten and

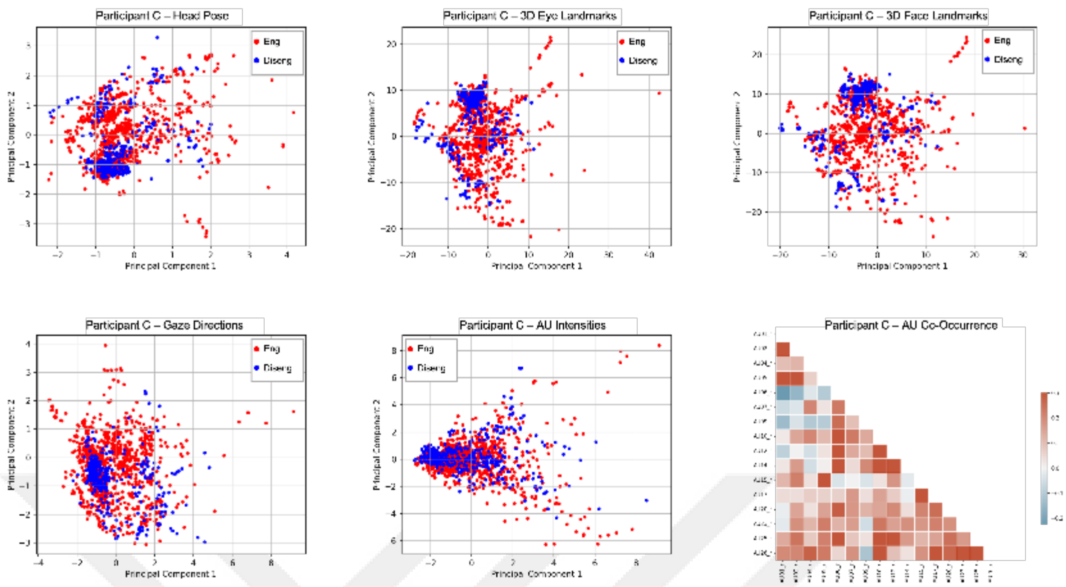


Figure 6.4: **Example PCA analysis of Participant C**: The figure shows the challenging nature of the data as there is no principle component that carries significant information. (Best for electronic distribution.)

Hinton, 2008, McInnes et al., 2018]. They are popular methods for visualizing high-dimensional space, but their non-linear nature can result in misleading output. I believe an effective way to explore this high-dimensionality space is to allow readers to explore the data with their custom parameters, which they can find a balance between local and global structure. So, I published the feature sets in *Embedding Projector*-friendly format (on Google Drive). Tensorflow's Embedding Projector [Tensorflow, 2022] is a web application to visualize high-dimensional format using PCA, t-SNE, or UMAP algorithms interactively. Through the Embedding projector, researchers can play with the dimensionality-reduction parameters.

6.2.4 Classification using Individual and Combined Features

In the classifier performance evaluation, I calculated the weighted F1-score metric using 80-20 split dataset. The F1 score is a harmonic mean of precision and recall. The weighted F1-score calculates scores for each label and finds their weighted average, which provides a more informed decision when data is imbalanced.

Classification using OpenFace Features: Table 6.3 reports the weighted F1 scores for five OpenFace features and selected combined features that I hypothesized combination could yield better accuracy: 3D Eye Landmark, 3D Face Landmark, Gaze Directions, Head Pose, and AU Intensities.

		LR	KNN	RBF-SVM	RF	BT
B	3D Eye Lndm	0.418	0.576	0.339	0.301	0.410
	3D Face Lndm	0.365	0.577	0.391	0.316	0.479
	Gaze Directions	0.294	0.384	0.385	0.310	0.384
	Head Pose	0.287	0.555	0.585	0.316	0.507
	AU Intensities	0.331	0.556	0.394	0.296	0.479
	All-OpenFace	0.418	0.576	0.338	0.306	0.501
C	3D Eye Lndm	0.399	0.596	0.515	0.440	0.526
	3D Face Lndm	0.468	0.686	0.564	0.448	0.597
	Gaze Directions	0.294	0.496	0.467	0.379	0.453
	Head Pose	0.370	0.688	0.659	0.452	0.584
	AU Intensities	0.318	0.609	0.479	0.318	0.511
	All-OpenFace	0.516	0.684	0.354	0.445	0.606
Y	3D Eye Lndm	0.472	0.558	0.390	0.488	0.525
	3D Face Lndm	0.499	0.662	0.446	0.504	0.591
	Gaze Directions	0.380	0.440	0.459	0.414	0.448
	Head Pose	0.411	0.648	0.631	0.518	0.524
	AU Intensities	0.390	0.589	0.518	0.432	0.517
	All-OpenFace	0.517	0.645	0.270	0.508	0.597

Table 6.3: **F1 Scores of OpenFace classifiers**: The classifiers are trained on individual OpenFace features (3D Eye and Face Landmarks, Gaze Directions, Head Pose, AU Intensities) and combination of all features for each participant. B, C, and Y are different participants.

Classification using OpenPose: Table 6.4 reports the weighted F1 scores for each participant's OpenPose joint keypoints. OpenPose returns twenty-five keypoint positions for each person in three axes.

	LR	KNN	RBF-SVM	RF	BT
P1	0.363	0.492	0.370	0.267	0.510
P2	0.332	0.461	0.363	0.264	0.459
P3	0.349	0.463	0.388	0.297	0.465
P4	0.367	0.517	0.461	0.281	0.524
All Features	0.474	0.439	0.324	0.256	0.517

Table 6.4: **F1 Scores of OpenFace classifiers**: Each OpenPose feature contains twenty-five keypoints in 3D world for each participant.

6.2.5 InterpretML Results

I run EBM and LIME on SVM-RBF for five feature sets for each participant similar to the classifier experiments. LIME and EBM can only run on binarized datasets, so I created an additional engaged-disengaged labels for binary classification task. In these experiments, I observed that action unit scores and head position information have the highest impact on the model at the individual level. An overall sample analysis of explainability scores looks like the rules in Table 6.5.

Researchers can access the full interpretability reports and generate their interpretability reports from Our Jupyter Notebooks Link.

6.3 Interpreting the Results of First Batch Analysis

Although engagement is a multi-component term, participants evaluated themselves using a single score in these studies. Throughout the annotation process, I observed that participants could score their engagement confidently after watching themselves for five minutes. But, compressing this multimodal definition into a single score brought an information loss, which resulted in various challenges in the data analysis. In this section, I thematically categorized the analysis patterns, opportunities, challenges and limitations.

6.3.1 Common Patterns in Engagement Levels

Engaging patterns: Our activity flow consisted of two parts where participants first learned to use creative coding applications following Youtube tutorials, then they completed a tangible activity that they followed using an online tutorial. While trying to complete the hands-on task, the participants used tangible materials and a tablet computer. During this interaction, the disengagement/engagement ratio of this part becomes low compared to the video-watching activity. This increase in engagement also resonates with the previous research in tangible interaction research. I promote adding more learning activity types, such as peer-to-peer learning and flipped classrooms, to increase the variety in the dataset and help educators interpret the patterns in different learning tasks.

Disengaging patterns: I can list two main patterns for disengagement: (1) When the instructor in the video started to talk about future work, participants tended to decrease their engagement scores. This situation might also occur since the timing of these conversations generally happens at the end of the lecture time. (2) When participants faced a technical failure, they did not feel disengaged but set their engagement to a moderate level. Listing these repeating patterns in engagement annotation can help educators to plan their lectures accordingly.

Outliers: Through the evaluations, I also faced unpredictable annotations. For example, when students were laughing and enjoying, one can expect to label it as an engaging moment. However, the engagement checklist also considers cognitive and behavioral engagement; thus, these enjoyable moments should relate to the course topic. However, I know that these moments help students keep their engagement level long-term. So, labeling and recognizing these moments are challenging both for ML models and teachers. Most of the false positive examples in the dataset consist of these moments.

6.3.2 Analyzing Unreliability in Self-Evaluation Process

Through the learning activities in the data collection process, I asked participants questions about the video content to check if their engagement levels and correct answers would match. The main goal of asking these questions was to see if their self-evaluation engagement score was in parallel with their answers. I aimed to achieve a reliability check with this kind of mechanism. After collecting the engagement level annotations, I checked the engagement levels one minute before and after the timestamp of this question.

The first question was, “What is p5.js?”. Two participants answered incorrectly. These students also labeled their engagement scores as negative. I also observed that the group activity model could also predict their engagement level correctly. But the second question was tricky. I asked, “Which platform can you use to code p5.js?” Only five of them answered correctly. They selected the platform that they used in the activities. But, the correct answer required choosing multiple platforms. In the evaluation part, most of them labeled themselves as ‘engaged,’ which indicates that most of the participants tend to label their engagement with their overall behavior rather than giving careful attention to the lecture.

Although this test can only be used for an exploratory and limited experiment for reliability, future researchers can follow a similar process to understand the reliability of their data collection process.

6.3.3 Interpreting Observable Features of First Batch

Our early experiments involving OpenFace and OpenPose features revealed that 3D Face, Eye Landmarks, and HeadPose features significantly improved accuracy compared to other facial, pose, and voice features. Explainability experiments also demonstrated that the combination of these features had the highest F1 score compared to other individual and combined features. Although gaze directions can reveal insights about shared attention between the students, the least successful models were trained with gaze directions. Previous research demonstrated that mutually shared attention could reveal some insights for classroom engagements. Still, I observed that gaze direction could also falsely guide the models when students are just in a mind-wandering state.

Through the analysis, I also check if the data features resonate with the literature. I expected to achieve the best results by either using AU intensities or combining AU intensities with other features, as the previous research could effectively predict affective states using action units. For example, the activation of AU4, AU7 and AU12 can indicate “Boredom”, and AU1, AU4, AU7 and AU12 can reveal “Confusion.” [Dewan et al., 2019]. But, when I checked if the students annotated ”disengaged” when the action units were active for the ”boredom” state, I could not achieve any significant result. To check the OpenFace's reliability in predicting action units, I also conducted a random frame check where I picked random frames and manually checked if OpenFace produced sound AU intensity values. In this examination, I concluded that OpenFace produces reasonable AU intensities, but I realized that predicting an engagement score using a single observation is a challenging task. For example, while they were eating some snacks while watching the video or looking serious while trying to solve a task on a computer, AU intensities were producing the same scores when they were disengaged.

Head pose, to my surprise, had a significant impact on all classification experiments. When I analyzed the feature importance reports, I observed that y-axis values are the main components. While watching the videos, I also observed that the participants (especially participant B) tended to node while listening carefully. I can also generalize this outcome intuitively and say that nodding and similar head pose behaviors can be key components of engagement analysis.

Before the analysis, I expected to see the major contribution of OpenPose features would be the change in the hand movements, head positions, and the distance between the participants. At the beginning of the analysis, I were planning to remove the lower body keypoints, as most of the points are not seen but predicted by the model. But, I decided to keep it, as participants stand up, talk to each other, or leave the class, which shows the change in the engagement levels. In the explainability analysis, I see that these keypoints had a major impact on EBM's decisions. So, future work should include either better keypoint prediction of occluded leg movements or increase the camera field of view to see the whole body.

6.3.4 Effective Data Analysis Practices

Combining features based on feature importance: Combining features from different knowledge sources can enhance classification accuracy significantly, but it can also cause overfitting. In the individual classification tasks, I observed that **3D Face Landmarks** and **Head Position** information are the main indicators of successful engagement level classification tasks, and combining them resulted in the highest scores for both classification and explainability experiments. Although combining features can improve the classifier performance significantly, simply adding all features together does not make the expected impact, as the classification algorithms struggle to find a converging path. So, while combining the features, I need to select relatively small-sized features that show high orthogonality with each other.

Crafting new features: I also tested creating additional features that might also be relevant to real-life situations. For example, I hypothesized that when students become closer in a classroom, it could be an indicator of engagement. So, I calculated the Euclidean distance between the participants' body keypoints and trained the classifiers with this closeness feature. But, the experiments did not yield the expected classification accuracy.

Adding more and more features: Increasing the future count with a limited dataset size results in weak accuracy scores, as converging to an optimal solution is a challenge. Selecting a subset of features is much more useful than feeding all networks to reduce the training and inference time. In the experiments, a 400-length feature vector took approximately spend 10-times more training time than a 200-length feature vector.

6.4 Implications

In a classroom environment, not every moment can be engaging. The goal is generally to minimize the long disengaged moments to keep the students' motivation high. So, it is vital to help teachers to analyze ML-based reports by combining multiple data sources. Based on the experience, I advise teachers to use ML-based engagement analysis when they already have some insights about students' characteristics, such as habits, likes, and dislikes. The ML models can only point out the engaged and disengaged moments based on some facial unit movements, body positions, and voice duration. This kind of guidance can be useful when students work together in a crowded classroom where the teacher cannot watch every group's behavior. However, in this scenario, the teacher still needs to take action, which can only be possible by re-evaluating the group dynamics and students' long-term behaviors.

The dataset and the early analysis proved themselves as an initial step towards the engagement level prediction and interpretation. The limitations such as representation diversity and annotation reliability limits the dataset, but the dataset has become a valuable source for experimenting ML techniques.

Component	**Rule**
Data Stats	The binarized engagement level of Session 2 have 4977 *Engaged* observations and 944 *Disengaged* observations.
Highest Score	The highest F1 score with the binarized dataset occurred when I fed 3D Face and Head Pose information, which yielded 0.84.
Action Units	Combination of action units had more impact than individual futures. Combinations with AU5 (upper lid raiser) had more impact than other action units.
Pose Features	All other features could show impact when only combined with other features. The individual features could not show a significant impact (<= 0.1). In all classifiers lower-body parts (RSmallToe, LSmallToe, RHell, etc.) had the highest impact on decision-making.
Morris Sensitivity [Iooss and Saltelli, 2017]	Similarly, Morris sensitivity values only yielded meaningful values for gaze direction and head pose features. In the head pose, I also observed the impact of y-axis values. The sensitivity values for pose features did not yield meaningful values (could not converge), which indicates the individual impact of features is very low.

Table 6.5: **A sample analysis of LIME feature values**: Interpreting a different model using LIME can favor different components. So, for different models at different inference times these components might change.

Chapter 7

PREDICTING ENGAGEMENT: DEEP LEARNING MODELS FOR CLASSROOM ENGAGEMENT

Notes to Practitioners:

- I developed two main model archictectures differing in the input modalities: (1) The MobileNet-with-MaxOut-layer architecture takes face images and achieves up to 85% test accuracy and (2) The MoViNetA4-powered model takes 10-seconds video clips and yields 71% test accuracy. I released the Jupyter notebooks in our Github repo.
- The model notebooks are open-source. I believe the dataset and models can accelerate the adaptation of recent deep learning models in exploring interpretable features to improve the current state of 21st-century learning environments.

Although engagement recognition from video has gained community interest in the recent years, limited models are avaiable for baseline comparison. Most of the existing public models and datasets are bounded to online learning setup, where they represent engagement using face-areas considering the affective states. These models do not consider the multimodal aspect of engagement. Limited studies consider these aspects, but they are not publicly available and their data format limits the capture of multi-modal engagement. So, there is no single task or baseline (e.g. BLEU [Papineni et al., 2002] in *Natural Language Processing* , ImageNet [Deng et al., 2009] in *Computer Vision*) that researchers in the field can compare and demonstrate success in engagement recognition in the wild. Although classroom engagement detection datasets and ML models are limited in number, the affective computing area aims to solve similar challenges. Researchers in video action classification, optical flow, and next-frame prediction areas can bring precious knowledge into classroom engagement prediction tasks.

After completing the data collection and cleaning steps, I developed baseline deep learning architectures to classify engagement levels in two formats: Image-based and video-based. Previous research demonstrated that using mid-level features using pre-trained networks is an effective method in video understanding [Acar et al., 2017]. Following this insight, I tested different mid-level representations and feature sets in our baseline architectures. In these experiments, I tested sequential learning architectures (GRUs and LSTMs) architectures with different feature representations (VGGFace, MobileNet-v2). I also tested integration MaxOut [Goodfellow et al., 2013] activation, as this method proved itself in different tasks with noisy or unreliable input.

This chapter introduces two new classification architectures that use different input modalities. The first set of architectures takes 320x320px size image inputs that contain cropped and centered face areas with individual engagement levels as labels. The second set of architectures takes 1280x720px size 10-second video clip inputs with aggregate group engagement levels as labels.

For all experiments, I splited the data into 80% training (20% of it is validation) - 20% test data.

7.1 Models using Centered and Cropped Face-Area Input

These models make predictions using 320x320px images, which contain cropped and centered face areas in the center of the image. I present two baseline architectures that test different pre-trained weight initialization and activation functions: (1) The first architecture tests two pre-trained weight initializations, VGG19-Face and MobileNet-v2. (2) The second architecture uses the MobileNet-v2-based model with MaxOut [Wu et al., 2018] and ReLU activation functions.

7.1.1 Experiment 1: VGG19-Face vs. MobileNet-v2 for pre-trained weight initialization

On top of pre-trained models (VGG19-Face[1] and MobileNet-v2[2]), I trained a densely connected NN with ReLU activation. The network uses sparse categorical cross entropy with Adam optimizer (lr = 0.0001). I trained the network for 50 epochs. Then, I also

[1]`https://github.com/rcmalli/keras-vggface`

[2]`https://tfhub.dev/google/imagenet/mobilenet_v2_100_224/classification/5`

tested fine-tuning the pre-trained weights to further improve the model performance. In the fine-tuning step, I used the RMSProp optimizer to restrict the oscillations that might occur in the gradient descent. I trained the network with the fine-tuning step with another 50 epochs. Using MobileNet-v2 as pre-trained weight initialization demonstrated better performance and accuracy for each participant's engagement classification. Thus, I tested the second model with only MobileNet weights.

7.1.2 Experiment 2: MaxOut Dense Layer vs. ReLU Dense Layer for Prediction Head

As I elaborated in Section 6.1 (Data Collection), the self-evaluation labels can be unreliable and noisy. If the noise and unreliable labels are not handled carefully, the network can learn a biased result. In the first set of experiments, the arhitectures use the Rectified Linear Unit (ReLU) activation that separates signals by a threshold to determine the activation of one neuron. When the neuron is not active, it returns 0. However, this mechanism can result in the loss of useful information especially for the initial convolution layers.

The Maxout function can approximate any convex function when there are enough hidden neurons [Goodfellow et al., 2013]. The Maxout operation can serve as a local feature selector for CNN architectures [Wu et al., 2018]. It is able to pick the best feature at each location, learned by different filters, by generating binary gradients that either activate or suppress a neuron during backpropagation. By using Maxout in a convolutional neural network (CNN), we can achieve a condensed representation while keeping the gradients of the Maxout layers sparse. The sparsity of these gradients has two benefits. Firstly, during the training phase, stochastic gradient descent (SGD) only affects the neurons that contribute to the response variables. Secondly, when extracting features for testing, Maxout can activate the most competitive nodes from previous convolutional layers by selecting the maximum value from two feature maps. This feature selection capability of Maxout also aids in generating sparse connections [Wu et al., 2018].

Following these insights from the existing research, I included MaxOut activation that showed promising results for noisy labels in previous face recognition tasks . For the MobileNet-v2 model, I tested ReLu activation and MaxOut activation for the prediction heads with a similar implementation to the LightCNN network [Wu et al., 2018]. The

MaxOut implementation, as expected, demonstrated better accuracy.

7.1.3 Results

The VGG19-Face is specifically trained to recognize faces, so I expected to observe a better weight initialization when I started our model training with VGG19-Face. However, MobileNet-v2 demonstrated significantly better performance in validation and test accuracies. So, I can deduce that MobileNet-v2 is a better feature extractor compared to VGG19-Face, even when the image area mainly contains faces. Thus, I continued with MobileNet-v2 as a feature extractor while experimenting with activation layers. In this experiment, MaxOut implementation demonstrated better accuracy as it reduced the possibility of overfitting and showed better accuracy in unreliable data. Table 7.1 reports the validation accuracy before and after fine-tuning and test accuracy of the final model for the MobileNet-v2 feature extractor with MaxOut activation, which performed the best accuracies among all experiments for the centered-face input.

Session	Person	Val-Acc	Val-Acc-Fine-Tune	Test-Acc
4	1	0.39	0.43	0.50
4	2	0.60	0.62	0.56
4	3	0.54	0.53	0.41
4	4	0.60	0.78	0.91
4	5	0.60	0.66	0.72
5	1	0.51	0.71	0.69
5	2	0.60	0.74	0.81
5	3	0.44	0.49	0.50
5	4	0.48	0.57	0.53
5	5	0.58	0.78	0.59
6	1	0.38	0.46	0.47
6	2	0.39	0.54	0.41
6	3	0.57	0.47	0.47
6	4	0.66	0.72	0.78
6	5	0.83	0.82	0.84
7	1	0.64	0.73	0.66

7	2	0.48	0.53	0.63
7	3	0.72	0.78	0.78
7	4	0.47	0.77	0.72
7	5	0.69	0.85	0.78
8	1	0.43	0.44	0.41
8	2	0.37	0.49	0.60
8	3	0.45	0.45	0.59
8	4	0.46	0.49	0.53
8	5	0.43	0.46	0.53

Table 7.1: **Face-based recognition results:** Validation and test accuracies of the MobileNet-v2 feature extractor with MaxOut activation for each participant.

7.2 Models using Video Input

The input of these models are 10-second long video clips that have 1280x720 resolution. In the data preparation step, I used a sliding window approach with a 2-second padding. Figure 7.1 illustrates this sliding windows approach. This means that I divided the recordings into multiple 10-second intervals, by moving 10-sec intervals by 2 seconds added at the end of each interval. Intuitively, we can say that if we select a random frame label, the 10-second interval (5 seconds before and 5 seconds after) has the same or similar label with the frame label. So, 10-second intervals that carries labels of their center frames. This way, I obtained 25K video clips that have 2 seconds of padding.

Using this data, I trained two different architectures with different base feature extractors: (1) The first experiment tests LSTM-based models with MobileNet-v2 and VGG19Face feature extractors. (2) The second experiment utilizes two different MoViNet architectures (A0 and A4) as the base video prediction model and applies fine-tuning to learn from few videos.

Figure 7.1: **Sliding Window Approach in Data Preperation**: I created 10 second long chuncks for model training by sliding the window by 2 frames. (FPS = 1, 2 seconds = 2 frames)

7.2.1 Experiment 1: RNN with CNN Feature Extractors

Our first model uses CNN-based models (VGG19Face and MobileNet-v2) that I previously utilized in face-area-feeding networks to LSTM-based network as seen in Figure 7.2. The experiment run for twenty epochs with the batch size 60 (60 video clips). The network uses sparse categorical cross entropy with Adam optimizer (lr = 0.0001).

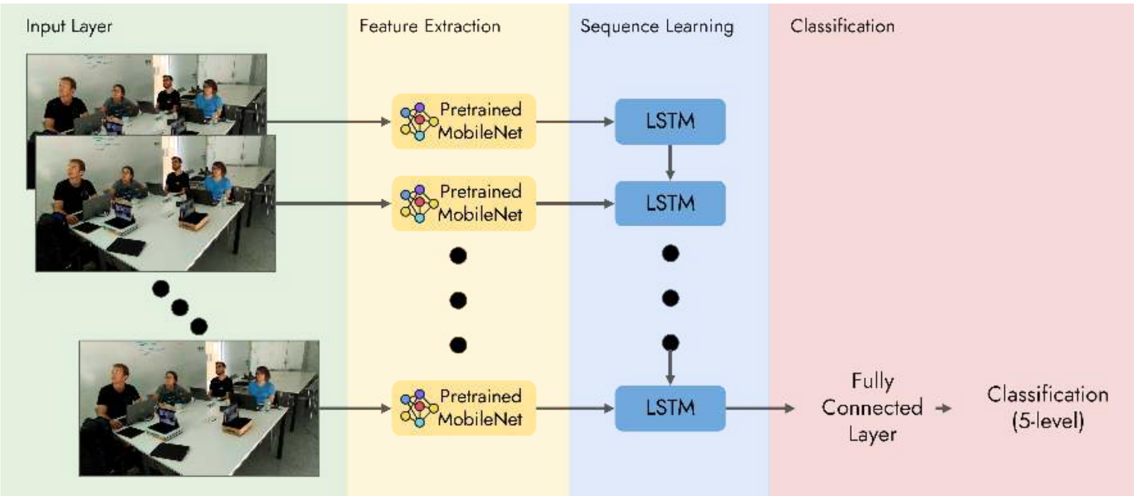


Figure 7.2: **Face-based engagement recognition model**: An overview of sequence-based learning using MobileNet-v2 features of individual frames.

7.2.2 Experiment 2: MoviNet A0 and A4 based Transfer Learning

Although CNN-RNN models have been dominantly used in video prediction models, a 2D-frame-based classifier lacks representation of temporal context. Tran et al. demonstrated that 3D convolutions have better performance in capturing spatiotemporal information compared to previous models [Tran et al., 2017]. In the second model, I used a descendant of a 3D convolution-based architecture, MoViNet. MoViNet introduces (2+1)D convolution approach in the video context. This convolution mechanism utilizes 2D convolutions and 1D temporal convolutions that mimics 3D convolutions, but in an efficient way. I used MoViNet A0 and A4 base models, which showed promising results on the Kinetics-600 classification task [Kondratyuk et al., 2021]. On top of MoViNet models, I trained a GRU-based classification head. The experiments run for twenty epochs with the batch size 20 (20 video clips). The network uses sparse categorical cross entropy with Adam optimizer (lr = 0.0001).

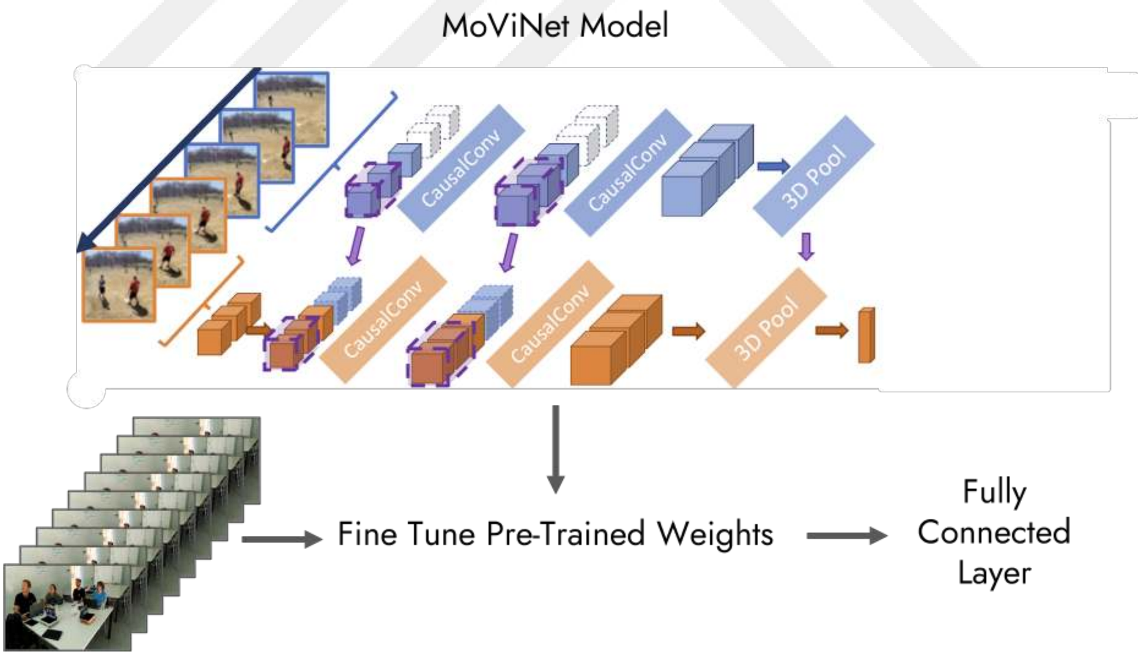


Figure 7.3: **Video-based engagement recognition model**: The high-level presentation of MoViNet-based classification using fine-tuning of MoViNet models. MoviNet architecture is obtained from [Kondratyuk et al., 2021].

7.2.3 Results

CNN-LSTM architecture yielded poor accuracy, compared to MoViNet-A0 and A4 models. Table 7.2 shows the validation and test accuracy results of these architectures. In the CNN-LSTM architecture VGG19Face and MobiNet showed similar performance in feature representation. Overall, the 2D-frame based representation could not yield any meaningful results. However, fine-tuning MoViNets yielded promising results. MoViNet-A4 base classifier could achive 0.71 accuracy on test set.

Architecture	**Val-Acc**	**Test-Acc**
CNN-LSTM (MobileNet-v2)	0.46	0.41
CNN-LSTM (VGG)	0.48	0.41
MoViNet (A0)	0.71	0.63
MoViNet (A4)	0.76	0.71

Table 7.2: **Video-based recognition results:** Validation and test accuracies of the different video action classification model architectures.

7.3 Implications

Our experiments demonstrated that carefully selecting pre-trained feature extractors in a transfer learning approach is the most important factor in achieving good accuracy. Currently, the most effective and high-accuracy models are run on top of MobileNet-v2 (for face-area-fed models) and MoViNet (group-activity-video-fed models) architectures. I encourage researchers to test different feature extractors and mid-level representations that could increase model accuracy significantly.

In addition, the dataset is limited in terms of generalizing the capabilities of the models. As I elaborated in more detail in Section 6.1.5, the dataset is collected in a university setting where participants are from the same department. This limits the affective, cultural, behavioral, cognitive, and pedagogical representation of the model. So, the adopters and researchers should investigate the ethical considerations related to representability and bias.

Chapter 8

INTERACTIVE DASHBOARD

Notes to Practitioners:

- The interactive dashboard is available as an Observable notebook to accelerate the customization and adaptation by practitioners. (`https://observablehq.com/d/21637c01efc12a65`)

Related Publications:

- Developing a Multimodal Classroom Engagement Analysis Dashboard for Higher-Education Students - `https://doi.org/10.1145/3593240`

Developing learning analytics dashboards (LADs) is a growing research interest as online learning tools have become more accessible in K-12 and higher education settings. The final stage of my end-to-end ML pipeline development for classroom engagement prediction was the development of an interactive dashboard to help students and teachers interpret the classroom engagement prediction of the system. The dashboard eventually resulted in displaying engagement logs in a row-by-row interactive structure. Using this dashboard, users can view their engagement over time and discover their learning/teaching patterns. In the development process, I conducted user studies with undergraduate and graduate-level participants to obtain feedback on the dashboard design. This chapter (**??**) presents a student-centric, open-source, and easy-to-access *dashboard* that both educators and developers can customize by their needs. (**??**) lists user insights and *design takeaways* for LADs and point at future directions for classroom engagement dashboards.

Making classroom engagement visible can help students to understand why and where they failed to grasp the concept and how they can succeed following their more engaging moments. Teachers can analyze their classrooms' engagement through different activities and manage their classroom flow accordingly. Considering these possible outcomes, I designed a dashboard that displays the engagement score with a cognitive self-evaluation assessment. I designed the dashboard to *motivate students to inquire about their learning habits* and *support teachers in thoroughly gauging classroom engagement.*

The dashboard contains four main components: (1) Information Area, (2) Engagement Score Chart, (3) Score Prediction Model's Rules, and (4) Cognitive Engagement Quiz Area. The *information area* describes the learning task. *Engagement score chart* visualizes the scores predicted by the baseline video classifier (Section **??**) for classroom engagement. *Engagement Score Chart* informs participants about the prediction rules to help users understand the underlying reasons. The video-based prediction model predicts engagement scores based on observable features of affective and behavioral engagement. To develop a dashboard that can help students self-evaluate themselves by considering their cognitive engagement, *Cognitive Engagement Quiz Area* generates a quiz that assesses students' confidence automatically based on the description in the information area.

After the initial wireframe prototype of the dashboard, I dynamically developed the dashboard on Observable through iterative user studies. Four undergraduate students and three Ph.D. candidates with teaching expertise joined the respective user studies. Throughout these user studies, I inquired about the experience of the dashboard, its perceived benefits, drawbacks, and obtained participants' suggestions. I analyzed the data in a deductive thematic analysis approach [Braun and Clarke, 2019] using the dashboard design motives: (1) Presenting an easy-to-follow interface that can accelerate students' self-evaluation of their classroom engagement. (2) Creating a system for students to discover their learning habits and engagement patterns regularly and willingly. (3) Presenting an explainable model to increase transparency. I present what I have learned from users with respect to each motive and the resulting interactive dashboard.

8.1 Dashboard Design

Throughout the design process of the dashboard, I aimed three tangible outcomes:

1. Presenting an easy-to-follow interface that can accelerate students' self-evaluation considering affective, behavioral and cognitive levels of engagement. I wanted to provide features that support meaning-making and reflection on the data.

2. Creating a system for students to discover their learning habits and engagement patterns regularly and willingly.

Problem Def. Component	**Explanation**
Stakeholders	Students, and Teachers. Our current research is not involving other education stakeholders like parents, schools, policy-makers, or NGOs.
Data Sources	Teacher provides a summary description for each learning task. The camera recordings of group activities is fed to the score prediction model, which yields the classroom engagement prediction scores. An external blackbox explainer like LIME yields the interpretability rules.
Data Logging	Currently, logging requires manual labor to run the model and establish the result score CSV. But, the automated process that starts with the camera recording is technologically feasible and possible.
Features of the learning setting	Our research considers only higher-education settings. But, I aim for generalizable results for K-12 education.
Design for Evaluation	From the beginning, I planned to conduct a face-to-face evaluation with small groups.

Table 8.1: **Applying LATUX in the dashboard design:** The main elements of the problem definition.

3. Increasing the system transparency by providing an explainable model.

In the problem definition, I followed a similar template presented in LATUX (Learning Awareness Tools – User eXperience) [Martinez-Maldonado et al., 2015]. The problem statement in the design process follows their template to standardize the ideation and development process. Table 8.1 shows the summary of the problem definition with LATUX components.

8.1.1 Dashboard Components

Considering the problem definition and main motivations, I designed a four-component dashboard. The components are (1) The information area, (2) Engagement Score chart, (3) Rules explanation area, and (4) Cognitive analysis area. I shared the initial design prototype of the dashboard in Figure 8.1.

The Information Area: This area includes the title and description of the learning task. The teacher is responsible for entering the title and description information in this area.

Engagement Score Chart: This area presents the engagement score obtained from the ML model. It is a frame-based timeline that shows engagement scores for each frame. I also added functionality to open the ten-second video intervals from the video recordings to help students remember their most engaged and disengaged moments.

Rules Explanation Area: This area lists the rules of ML explainability models (e.g. LIME) in a human readable format. Normally, black box explainers like LIME return the activation points of the models and their relative scores, which are still unreadable for the end user. So, I automatically generated more understandable rules in text format. The For example, when LIME returns 150 activations with the highest mean absolute score (MAS), like "AU 4 ¿ 0.9234 and AU5 ¡ -0.643 - MAS: 0.3", I report this as *The system determines this score by creating more than a hundred rules. The system could not find one single significant rule. But, when the students have active "Brow Lowerer" and inactive "Upper Lid Raiser" action units, the system is predicted as an "Engaged" moment.* The generation is also a rule-based process that simply maps the features to their human-readable word formats.

Cognitive Analysis Area: In the multimodal analysis, the machine learning model determines the score based on affective and behavioral engagement checklist items. Determining cognitive engagement requires the self-assessment of the students. I aimed to introduce a small questionnaire in each report to help students assess their cognitive engagement.

The quiz in this area is automatically generated from the description in the Information Area. For example, if the description says, "Creating an abstract painting like Vasarely from scratch in TouchDesigner to explore the capabilities of the tool.", the system can generate a Likert scale question like "What is your level of confidence in creating an

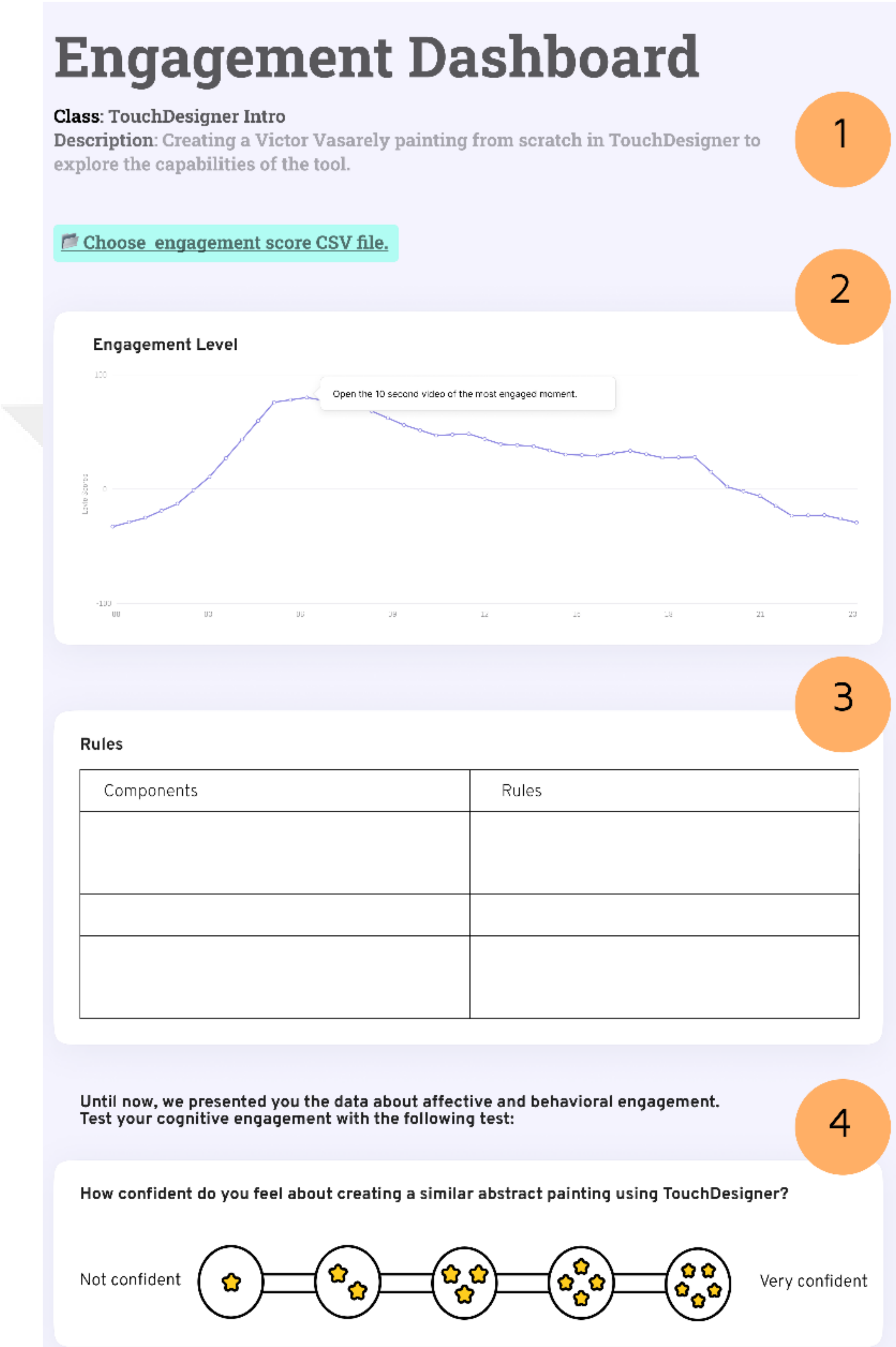


Figure 8.1: **The initial design of the dashboard**: Four components are numbered on the figure: (1) Information Area, (2) Engagement Score Chart, (3) Rules Explanation Area, (4) Cognitive Analysis Area.

abstract painting in TouchDesigner?" or "Mark your confidence level in using different capabilities of the tool." Currently, the system uses a rule-based parser to generate a question from the given nouns. But, it is also possible to fine-tune a language model to generate these cognitive questions from these small descriptions.

8.1.2 Software Development

I developed and served the dashboard on Observable[1], an online interactive development environment specifically designed for data-related scenarios. Our main goal in choosing Observable as the development platform was to dynamically change the views and interaction methods in the user studies. The platform choice also supported making the dashboard extendible and modular for researchers, educators, and end-users. The notebooks both present an interactive development environment and a good-looking interface that appeals to the end user. An experienced user with some web-development experience can also tweak the code easily and extend the interface for different use cases. Figure 8.2, 8.3, 8.4, and 8.5 shows the component rows from interactive Observable notebook. The dashboard is available at `https://observablehq.com/d/21637c01efc12a65`.

Figure 8.2: **Displaying Overall Engagement**: I displayed the overall engagement using a caret pointing on a discrete gradient bar. The chart implementation is an updated version of @duckyb's fever-chart

[1]`https://observablehq.com/`

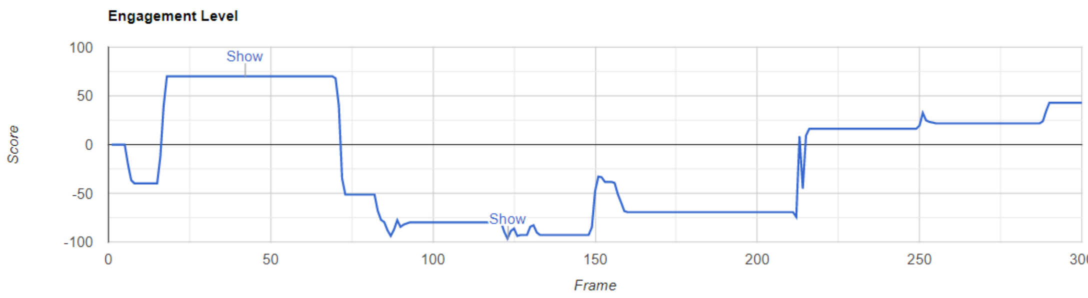
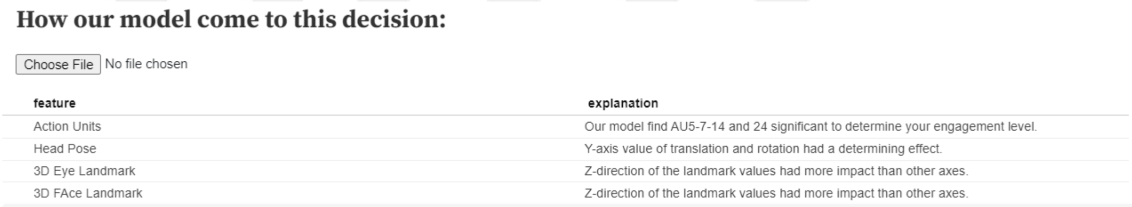


Figure 8.3: **Engagement Chart**: I used Google Charts to show display the engagement scores through time. When the user hovers on Show button, the chart displays corresponding ten-second video.



How our model come to this decision:

Choose File No file chosen

feature	explanation
Action Units	Our model find AU5-7-14 and 24 significant to determine your engagement level.
Head Pose	Y-axis value of translation and rotation had a determining effect.
3D Eye Landmark	Z-direction of the landmark values had more impact than other axes.
3D FAce Landmark	Z-direction of the landmark values had more impact than other axes.

Figure 8.4: **Rules Explanations**: Rules are extracted from the LIME algorithm and later simplified manually in the produced CSV. In the Observable implementation, I manually selected the rules file and displayed it on tabular format.

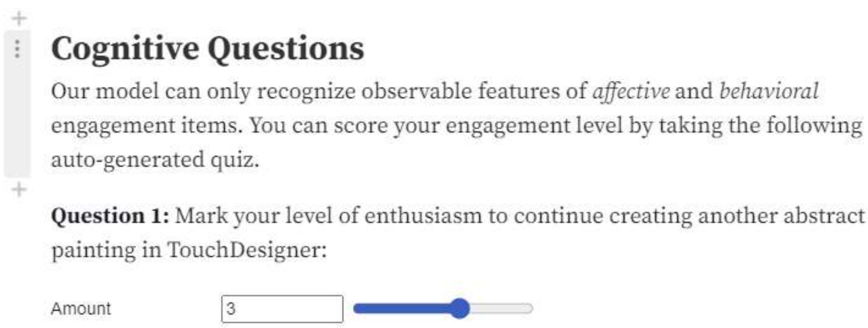


Figure 8.5: **Cognitive Analysis Test**: I showed two questions about their enthusiasm and confidence, based on the description of the current learning task. Participants used a slider to select their level between 0-5.

8.2 Study and Analysis

8.2.1 Participants and Setting

I conducted two respective user studies with the same participants that also attended the *Classroom Engagement Dataset* collection sessions. Both studies took place in the authors' university. The first user study was with four undergraduate students and the second one was with three Ph.D. candidates. The undergraduate students were recruited using University's student forum that explained the study aim and details. The Ph.D. candidates were specifically recruited from the *Interaction Design Lab* as I wanted to gain in-depth insights on the experience and obtain suggestions on the LAD. The participated Ph.D. candidates have more than three years of teaching experience. Therefore, the Ph.D. candidates were able to assess both the learning and teaching side of the classroom.

8.2.2 Study Design

The dashboard is a one-page interface with four components that consists of novel interactions. To understand the usability and assessment concerns, I conducted a qualitative study that involved talk-aloud sessions rather than a quantitative approach. I prioritized understanding the user's perspective, rather than measuring the statistics of the users' ability to learn and use the system.

In both sessions, I utilized the DATUS (Dashboard Assessment Usability Model) questionnaire as the basis of the discussions. DATUS combines well-known usability metrics with a focus on learning analytics literature, which provided us with a perfect survey to discuss around. The questionnaire also follows the ISO-9241-11 standards and groups its evaluation metrics into eight dimensions: Effectiveness, Efficiency, Satisfaction, Learnability, Accessibility, Recognizability, Aesthetics, and Operability. I shared the questionnaire in Appendix B.

I built the semi-structured discussions inspired by the DATUS questionnaire along these four inquires:

- **(RQ1)** How intuitive is using the dashboard and exploring the engagement data using the provided tools?
- **(RQ2)** How can I develop this dashboard prototype further to meet your expecta-

tions?

- **(RQ3)** How can the empirical results inform the design and software updates of the dashboard?

Note that due to the semi-structured nature of the discussions, the questions changed and extended. In addition, I asked additional questions to understand their current practice in learning analytics and asked if and how the dashboard would fit into it.

8.2.3 Procedure

Our studies roughly followed a three-step procedure (i) Discussing learning analytics habits, (ii) Introducing Dashboard Cards, (iii) Overall discussion.

In the first session, I met with four undergraduate participants. I showed the initial design prototype of the dashboard (Figure 8.1) and the participants explored it in a think-aloud fashion. I also asked the semi-structured questions listed earlier. In the meantime, I actively made some updates to the dashboard based on participants' suggestions and concerns in real-time. This way, participants had the chance to observe their ideas in-situ that provided a reflective discussion. In the second session with Ph.D. candidates, I presented the iterated version of the dashboard based on the first user study. Each study session lasted around one hour.

8.2.4 Measures and Analysis

The study data consisted of audio recordings of semi-structured interviews and the principal researcher's field notes. After familiarizing ourselves with the transcript of the recordings, I thematically grouped the statements with an external researcher [Braun and Clarke, 2019] in a deductive manner. The motives in the dashboard design guided the themes.

8.3 Results

Given that the user studies are conducted with two different groups, undergraduate students and Ph.D. candidates, the discussions differed. While the undergraduate students' dominantly talked about the possible benefits and drawbacks of the interface, the Ph.D. candidates lengthily discussed the visualization, and possible effects of scoring rules on teachers' and students' behaviors.

Below, I presented the findings in line with the design motivations, by summarizing what I have learned from the user studies to help address the motives more effectively.

8.3.1 Learning Analytics Habits

I summarized participant's personal analytics habits and listed regularly used analytics interfaces to give an overall view on their data literacy skills.

- *Physical note-taking:* P12 stated that she generally uses her personal notebook and takes notes about task completion. After she shared this habit, all undergraduate participants also noted that they do this kind of note-taking by not realizing it is a type of learning analytics. The participants in the second session also use Notion to follow their tasks, but they all use notebooks as a quick tool to check regularly.
- *Regularly used learning apps:* Participant's were not actively using analytics software or checking integrated analytics dashboards. But, they were all familiar with the concept, as they used Blackboard in their classes. Undergrad participant's noted that they only use Blackboard to follow course content and keep track of their grades. They also reported to follow Coursera, edX, and TreeHouse courses, but they only viewed their general learning progress available on the home page.
- *Regularly used external analytics apps:* As part of the interest, I asked participant's analytics app use- beyond learning context. Participants mentioned using ZeppApp (i.e., a sleep pattern app), Youtube Analytics and screen time reports. All participants emphasized that they like analytics that combine various data sources (e.g., location, duration, etc.) in a comprehensible manner. To illustrate, one participant mentioned that he uses Youtube Analytics regularly, but he strongly dislikes it as it provides many hard-to-read charts all at once. However, they like tracking apps (i.e. fitness or sleeping) as these apps are producing simple, understandable numeric representations and visualizations.

8.3.2 Providing a Space for Self-Evaluation

Our initial motivation was developing an *easy-to-follow interface that can accelerate students' self-evaluation of their classroom engagement.* To that end, I wanted to provide features that support meaning-making and reflection on the data. Our findings revealed that presenting data in an easy-to-grasp format while leaving room for interpretation was important in this regard.

- *Overall score as preferred data format:* Seeing a more overall score by combining with other learning data could make sense by using it. In both studies, participants gave examples from sleeping apps. For example, one sleeping tracker app calculates an overall score by looking at all patterns. It makes little sense in the beginning, but when getting used to it, just looking at it can show quick interpretable information for the user.

- *Visiting the peak moments for interpretation and learning strategy:* Participants valued seeing the small sequences of the most and least engaged moments. These moments were noted as a helpful feature to help understand the models' engagement score. Participants also noted that they would visit these moments to see if their engagement was related to the class itself, or were they distracted/entertained by something else. Therefore, this feature was perceived as a double check on their score's accuracy. Also, disengaged moments were considered keys for exam prep: "I wish I had something like it in my undergraduate years so that I could just check my disengaged moments and focus on studying them".

- *Recommendations on boosting one's cognitive engagement:* I avoided integrating personalized recommendations in the design of the dashboard. But, undergraduate participants stated that they would like recommendations. "I would like to see personalized suggestions on how to best follow the courses, similar to sleeping pattern suggestions in sleep tracker apps."

- *Valuing individual student engagement:* One Ph.D candidate stated that the dashboard would allow her to more effectively evaluate engagement based on each student: "Mostly, I assess students in a standardized format. Ill, I actually do care for individual differences in participation. For example not everyone who talks is cognitively engaged or the silent student could be the one most minds-on with the content. Yet when the classroom size grows, or there's a group work, these individual differences get a bit harder to see... Watching these videos and evaluating them with a machine learning model can enhance the way I interpret student engagement."

- *Testing more compact ways to present the data:* In the second session, participants asked if I can create a more dense visualization that can immediately show the overall engagement. Using the five-level engagement definition, I can also present the engagement through time in a more compact way as seen in Figure 8.7. In the studies, I demonstrated this representation on Observable. They stated that, in the compact version, they were looking at the points where the colors change, which is a more visible indicator and eases the evaluation process. P22 stated that, "... with this representation, I could investigate the red (disengaged) areas in more detail."

8.3.3 Visiting the System Willingly

Our second goal in the dashboard design was *creating a system for students to discover their learning habits and engagement patterns regularly and willingly.* I present participants' comments and suggestions in this regard.

- *Using extrinsic motivation to boost intrinsic motivation:* All participants noted that only a small percentage of the classroom would self-evaluate themselves. They emphasized that completing the cognitive engagement quiz should be mandatory in the beginning. One participant stated that "I wouldn't do any quiz if the teacher wouldn't make it mandatory. But, once I start controlling the system regularly and see my positive progress, I would like to visit the dashboard regularly." This touches upon sources of motivation, a topic widely discussed in education [Shabaninejad et al., 2022]. Participant insights suggest that doing something for an instrumental goal (e.g., grade) known as extrinsic motivation, could initiate intrinsic motivation (e.g., checking the system for its own sake) in the students.

- *Sending engagement reports in intervals to maintain morale :* Participants noted that each class is different and it is not possible to always remain engaged. Therefore, class-by-class engagement reports were not favored. One participant asserted: "This (disengagement) report would upset me, say if I had a recent break-up... It would show me how I became really disengaged. It could cause stress rather than self-awareness." This led to a discussion on using intervals (e.g., every four-weeks) for sending engagement reports to students. This would account for the weekly differences.

8.3.4 Presenting a Transparent Evaluation

Here I present the findings aligned with the third motive while developing this interface: *Presenting an explainable model to increase transparency.*

- *Building trust through transparency:* All participants reported that they trusted the current model's scoring predictions because it provided the backstage of the results (i.e., the rules). The transparency was valued from a teacher perspective, one Ph.D. candidate noted: "How should I react when a student comes to us, saying that she is already trying hard, but the system keeps scoring her engagement low? I mean the reasons behind their scores should be clear enough for them to see." However, the way the rules of the classification system were presented were of critical importance, as I further eloborated in the following items.

- *The optimal transparency:* In the first session with undergrad students, presentation of rules was not viewed as an important feature of the dashboard. The participants mentioned that seeing the rules is interesting, but they would only look at it once. They suggested that integrating these rule explanations into video can be more engaging. In the second session with Ph.D candidates, this rule system was discussed more in depth. These participants noted that the detailed explanations on the model might be exploited by the students. To explain, if the model presented that gaze direction as a significant influencer on the score, it would cause students to track teachers movement and receive high engagement scores. However, it was also noted by participants that this 'mimicking engagement' might have a positive effect since engagement-mimicking behavior can result in real engagement.

- *Avoiding self-report on cognitive engagement:* While presenting rules is a way to introduce the inner workings of the scoring system, the app itself also gives participants a hint. To explain, the cognitive engagement score is obtained from self-evaluation. However, one participant stated that "I may not trust the cognitive engagement scoring ability of a system by just answering some "how confident" questions. Because maybe I am wrong and I do not actually know my comprehension level". Another participant noted that answering small and simple questions could be more reliable and would also enhance the learning progress.

8.3.5 What to present, when to present

Here, I present the findings on the ideal ways to present the engagement data.

- *Visibility of analytics' features:* While discussing participants' regularly using learning analytics apps, they noted that they don't know which platform has which feature. For example, Blackboard integrates many analytics, discussion and evaluation tools. But the participant's did not know this as they cannot easily find these tools. Therefore, it is fundamental to make these analytics visible in the dashboard. While making the analytics visible with reminders or straightforward homepage scores are the first ideas that come to mind, it pertains to some issues discussed below.

- *Reminders are stressors:* The participants didn't like the idea of a regular engagement information reminder. It was asserted as a stressor rather than an informative feature. When I discussed presenting engagement scores not as reminders, but as information on the dashboard, the participants noted similar concerns. Showing the tasks and scores on the home page when they first opened the system was perceived to make them feel stressed. This point is with the following two topics on presentation type that suggests how to combat the perception of data as a 'stressors'.

- *Avoiding detailed numeric presentations:* All participants agreed upon seeing really low values in the Overall Score area would cause disappointment, which would eventually lead to not visiting the interface. For example, if I use the current scoring system between -100 to 100, scoring "0" is moderate engagement, but it could upset the user. So, they stated that they would rather see a score between 0 to 100. So, instead of "0", they would see "50". In addition, some participants could not understand what a negative engagement stands for.

- *Visualization over metrics:* In line with the previous point, participants noted how they preferred abstract visualization rather than scores and metric-based charts. In the second session, participants offered a visualization system that can illustrate the relative distance to the high-engagement. They also stated that this kind of visualizations can make the system more playful and would benefit from incorporating gamification.

8.4 Revisiting the Design of the Dashboard

After grouping the user study results, and list the possible updates to the dashboard, I first re-visited the problem definition. Then, I updated the dashboard considering the requested reporting format for overall engagement prediction scores, its abstracted visualization, and visual and data-level abstraction consistency.

8.4.1 Revisiting the Problem Definition

Stakeholders

In the studies, I primarily focused on getting students' perspective, as this dashboard primarily focuses on increasing students' ability to self-evaluate their engagement. Yet, in the second session, comments of Ph.D. candidates also helped us to curate some considerations from a novice teacher perspective. Novice teachers can be one of the main beneficiaries of the dashboard to observe their students and understand their behavioral patterns. Considering the user study results, I defined "teachers" in a more comprehensive way based on their expertise level. In addition, the user study results revealed the students' expectations from teachers come in three-folds: (1) Ensuring everyone interprets the engagement levels clearly, (2) Helping students to evaluate their engagement considering all three aspects of engagement and (3) Moderating the classroom dynamics based on the results of engagement analysis.

Interpretability

In the problem definition, I determined the primary data sources as engagement scores and interpretability rules which were yielded by XAI algorithms like LIME. Our initial aim was to enhance the users' trust in the system. However, combining the dashboard with blackbox model explainers like LIME did not appear as a must-be feature in the studies. In contrast, the students were asking to combine the engagement scores with external personal data sources, such as wearable data, location information and assessment scores, which would result in more complex models with less explainability.

Considering the early results, I can make the deduction that presenting interpretable rules can help students to trust the system only when the presented *rules* are just shallow explanations. As reported earlier, when the rules become more detailed, students might start imitating the behaviors that the model favors. In addition, some concerns over interpretability put extra work on teachers' shoulders. For example, one question raised in the user studies was how teachers should react when a student comes to them, saying that s/he is already trying hard, but the system keeps scoring her/his engagement low. What should be the teacher's reaction? In this case, should students find favor in the camera's eyes or the teacher's eyes? Answering these questions requires a harmonic collaboration between teachers, students and the ML-powered dashboard, where all parties understand the fact that the model just produces some scores based on some patterns, and they should interpret these scores considering their overall engagement progress.

Dealing with motivation

I listed several concerns and challenges in the previous section. A possible unwanted outcome can be causing stress rather than self-awareness. The listed concerns are generalizable for all LADs that show longitudinal data, and should be carefully tracked by educators while using these analytics interfaces. Although limited research is conducted on dashboards' effect on affective states of the students, recent studies by Bennett [Bennett, 2018] and Muldner et al. [Muldner et al., 2015] and report by Sclater et al. [Sclater et al., 2016] demonstrated that most students in both K-12 and higher-education levels were motivated by seeing their data presented in a dashboard format, which led to positive behavior changes. Our user studies revealed that the disengaged moments should be carefully treated, as these may cause even more disengagement. It was also noted that disengagement might also stem from non-class related issues (i.e., going through a break-up). Therefore, it was suggested that presenting classroom engagement in intervals (e.g., every four weeks), in ambiguous visualizations over harsh metrics would be more favorable.

8.4.2 Planning the Next Iteration

After analysing and interpreting the results of the user studies, I re-designed some components of the dashboard. Figure 8.6 shows one row of the engagement prototype of the updated version of the interface. Following participants' suggestions, I focused on two main elements while updating the interface: (i) Visualization of class-level and overall engagement, and (ii) consistency of data representation.

1. *Representing Engagement:* The left side of the row in Figure 8.7 shows the new stacked bar timeline visualization which displays the categoric engagement levels in a compact form. During the video is playing, a small icon's (running person emoji) distance is changing relative to the engagement icon (books emoji).

2. *Keeping consistency:* In the previous version, the engagement chart and rules explanation was presenting a very detailed and numeric way of tracking engagement. Compared to the engagement score visualization, cognitive engagement was too shallow, which caused questioning the confidence of the system. In the current version, I included a small quiz related to the activity rather than asking questions about enthusiasm and confidence. Additionally, their overall engagement score can increase by answering these questions correctly, which unified the all interface elements in one representation.

Interactive Rows of a Digital Notebook

I designed the overall view similar to a interactive notebook-like structure. This idea is motivated by their physical notebook usage habits. Following their naturally developed physical notebook habits and suggested features, I designed this format to present the data in a more compact form. Recent research also suggests that integrating classroom data in a format that comes natural to students provides a more exciting interface for students' self-exploration [Pozdniakov et al., 2022, Fernandez Nieto et al., 2022]. I also integrated the rules in the overall view, by reminding students about how the model made

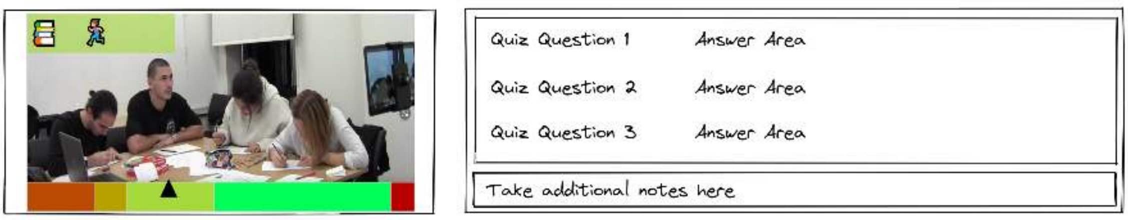


Figure 8.6: **A sample row of the journal style engagement logging**: The left side shows the video with engagement score timeline. The right side asks small auto-generated questions about the course and allow adding additional personal notes.

Figure 8.7: **Offered abstract presentation of engagement level**: In the upper-left corner, a colored area (the color is the current engagement category area) shows the distance to the high engagement via emojis. At the bottom, the current video timestamp moves through the engagement category stacked bar chart.

the decision in the video sections spontaneously.

Engagement Score Abstraction

I created a stacked bar chart to visualize the engagement categories through the lecture. I additionally added an abstract visualization of "engagement distance," where the "Running Human" emoji tries to reach the "Book Stack" emoji, which displays a gamified illustration of the engagement score. Figure 8.7 shows the new video-watching interface with these abstraction features.

Color Components

I used a traditional gradient from red to green to represent the five-level scale of engagement. Then, I used Viz Palette [Lu and Meeks, 2022] to check the color space for different types of color blindness. Figure 8.8 shows the finalized colors and their HEX values in a sample engagement stacked bar timeline chart.

8.5 Implications

Following the user studies, the considerations on analyzing students' learning analytics habits, providing a space for self-evaluation, boosting their intrinsic motivation, and presenting a transparent evaluation can benefit researchers from all disciplines that would

Figure 8.8: **Color components**: The timeline view of engagement categories in a stacked bar chart style. From green to red, the HEX values are: [”#b70000”, ”#bb4b00”, ”#b7a100”, ”#a5d93c”, ”#00ff58”]

like to design a learning analytics dashboard. From a broader perspective, I expect the research to be integrated into novice teacher education by providing an easy-to-use tool to give hints about groups’ engagements and help them build engagement-related conversations with students. I expect the models and dashboards to be a part of crowded classrooms as a helper tool to better evaluate overall engagement.

Chapter 9

DISCUSSION

A discussion on future paths of this research and building ethical and pedagogically-appropriate child-centered intelligent systems.

Throughout this dissertation, I have built consecutive products upon the related discussions and findings aroused out of the previous stages in an iterative approach. This chapter presents a more generalized discussion around challenges and open questions for developing intelligence in child-centered product development and the future paths of my research. First, I establish a set of questions that I have been asking in each stage of ML pipelines to make these processes more child-centered (Section 9.1). Then, I discuss the next steps in developing intelligent educational applications (Section 9.3). Following this discussion, I interpret the classroom engagement dataset findings to inform the possible future applications in the middle school domain. Finally, I listed the ethical considerations for using the research outcomes in classrooms for both researchers and practitioners (Section 9.2).

9.1 Developing ML Pipelines from a Child-Centered Perspective

Machine Learning pipelines stand for all the necessary steps to develop ML systems (e.g., developing a predictive keyboard) [Paleyes et al., 2023]. Although no standardization is defined, an ML practitioner can list the stages of a typical machine-learning pipeline in six main categories [Lavin et al., 2022]. Following these stages with a human-centered focus can help practitioners to build natural, intuitive-to-use, high-performance intelligent systems that also abide by ethical considerations. Throughout my Ph.D. studies, I both used pre-trained, ready-to-deploy ML models and developed new models starting from scratch to increase the affordability of programming education and help teachers manage the classroom environment more easily. In Table 9.1, I listed the steps and the corresponding questions in the pipeline steps to make these projects more child-centered, and

I summarized the two main ML Pipelines of my Ph.D. work to reveal how these questions arose in the development process.

9.1.1 Pipeline #1: Classroom Engagement Prediction

Problem Identification: Since teachers in crowded classrooms have challenges in tracking students' engagement levels, the problem definition has become, "How can I support teachers while determining the engagement level in classroom activities?" This problem requires input from students (a video, bio-data from smartwatches, etc.) that can reveal their engagement levels and a model development that can learn from the given input. In the literature review, I encountered that existing vision models have the capability to understand video actions and determine engagement levels. But I needed a new dataset for training a new engagement-level classifier since the previous datasets only considered facial features in online learning. So, I collected a new dataset and developed a baseline architecture.

Data Collection Ironically, I did not meet with the children to make the process more child-centered. I first collected data from university students. This way, I was able to publish the dataset in open-source format. The ability to open-source the dataset is one of the main reasons behind the fast-pacing development of AI. The university students could also self-evaluate their engagement level, which allowed us to collect engagement level labels in an effective way. After following an iterative process and making the data collection activities more reliable, I decided to collect data from middle school students to fine-tune the model and develop few-shot techniques. So, I did not collect any data until I made sure every step was suitable for children. I did collect a minimum amount of data to keep the well-being and privacy of students.

Exploratory Data Analysis: The main goal of the process was to learn the observable features that can contribute to learning to determine the minimum amount of data to collect. In this analysis, I applied dimensionality reduction, some basic classifiers, and feature engineering steps to observe the effect of observable features such as 3D face landmarks and body keypoints. After the analysis, if I could determine engagement by looking only at children's right hands, this could allow us to collect video recordings of right hands, which enables us to anonymize the data automatically.

Model Development: Since the dataset is new, I developed baseline architectures.

ML Pipeline Stages	Child-Centered Questions
Problem Identification	- Is the problem focusing on data collection, model development, or interaction? - Are there any child-centered use cases or guidelines for this problem?
Data Collection	- If there is an already existing dataset for the identified problem: How can a researcher update it to make it more child-centered? - If no dataset is available: What should be the steps for involving children in the data collection process? - If it is a supervised problem, how can a researcher make the process more "child-centered"?
Exploratory Analysis	- What should be the data analysis steps to check the child-centeredness of the dataset?
Model Development	- If it is a baseline architecture: Are your test cases representative enough to call the model "child-centered"? - If it is an improved version of the previous models: In which directions do your model improve the existing models from a "child-centered" view?
Ethics Considerations	- Can your data, model, or end-product interface potentially harm the privacy or well-being of children? If yes, how can you prevent it?
Interface Development	- How can a researcher involve children in making interface development more "child-centered"? What are the new challenges involving children? - What kind of new modalities does your interface use? Can children face challenges while using these new modalities?

Table 9.1: **Child-Centered ML Pipeline Questions**: Machine learning pipeline steps and the questions during these steps to make the development process more child-centered.

In the model development, I followed transfer learning techniques to learn better using a few data. I focused on the affordability of the final setup to run the model using an already available computer and a single camera. So, I used MoViNet as the main feature extractor, which previously demonstrated success in predicting actions in streaming videos. After developing the model, I also listed the model's limitations for future practitioners. Our model only represents group activities completed with tablet computers by students with good socio-economic status.

Ethics Considerations Bringing a camera into classrooms can reveal some concerns related to classroom surveillance. I designed the data pipeline and dashboard to aid students and teachers in observing engagement patterns and interpreting their engagement levels. Yet, one can be concerned about using the dashboard as a classroom surveillance tool, which can process personally-identifiable data that classifies behavior, attitudes, and preferences. Privacy concerns have been a prominent challenge in the learning analytics field. Williamson et al. list four emerging challenges while developing LADs [Williamson and Kizilcec, 2022]: (1) Protecting participants' privacy while also including enough demographic information, (2) Surveillance concerns, (3) Neglecting pedagogies that fall outside of the dominant narrative and (4) Making LADs maintainable in terms of software development. For each of these emerging issues, I actively communicated with researchers, teachers, and policymakers to make the dashboard more accessible and reliable for all.

Interface Development Students were active partners in the development process by actively guiding the interactive coding process. In the dashboard development, I used Observable[1] and coded it in the user studies where students comment in real-time. Using these interactive, modular notebooks or low/no-code platforms can increase the interactivity of the sessions.

9.1.2 Pipeline #2: Kart-ON Programming Environment

Problem Identification: The problem focuses on using existing vision-based ML models in a pedagogically-appropriate way. Using everyday objects as programming materials can help the abstraction capability (e.g., creating functions and variables) of tangible programming environments. However, this kind of setup requires scanning these objects

[1]`https://observablehq.com/`

using a mobile device throughout the programming activities. For example, when a student wants to represent a function that can draw a tennis ball, the student can use a real tennis ball to represent this function. But the question is *what are the system requirements to help children use these everyday materials in programming activities seamlessly?*

Data Collection: In this scenario, I did not need any new data collection to develop an object recognition system. But, the scenario required children to collect new data throughout their programming activities. So, I conducted a participatory/informative process so that I could gather exploratory information about their needs and habits [Sabuncuoglu and Sezgin, 2022a, Sabuncuoglu and Sezgin, 2022c]. In an iterative set of activities, I conducted three different activities that involved an increased level of programming activities (unplugged, Flappy Bird, Teachable Machine) in understanding their programming object creation ability.

Model Development: I used a ready-to-deploy model, MobileNetv3, from Tensorflow Hub[2] that was already engineered to run on mobile devices. I developed an on-device data pipeline that can take a few photos in Android smartphones and use it in programming activities [3].

Ethics Considerations: In this system, the training and inference take place on-device. So, I did not collect any data from children. I suggest all researchers consider on-device [Dhar et al., 2020] and federated learning [Kairouz et al., 2021] scenarios if they plan to actively update the ML model while preserving children's privacy.

Interface Development: In the development process, children obtained three roles: tester, informant, and participant [Druin, 2002a]. I had the opportunity to observe and learn new user-experience considerations for future iterations of the software. In the previous studies that involved "Teachable Machine," I observed that making the system's AI capabilities visible helped students solve problems much easier. For example, if the AI system cannot recognize the object due to different lighting conditions, the visible training data reminds students that AI only learn from one light condition. So, I did not avoid using AI terms like "data," "training," and "inference."

The child-centered questions of *Pipeline #1 and #2* can support researchers and practitioners in achieving the development of pedagogically and ethically sound intelligent

[2] `https://tfhub.dev/`

[3] `https://github.com/karton-project/karton-android`

systems. These questions are aroused from my experiences in developing intelligent educational applications. Yet, there is still room for improvement in the child-centered HCI field, and researchers should share their experiences to develop a more comprehensive guideline.

9.2 Future Directions for Research Outcomes to Accelerate the Classroom Use

Kart-ON's main goal is to make programming more available to all students, regardless of their socio-economic background. I believe the learning content of a curriculum should be designed to promote the learning approaches that every student in the country can apply and afford. Increasing the accessible, affordable, and playful practices in the design of tools and learning materials can accelerate the wide adoption of the curricula. Based on my observations and classroom experiences, I summarized the future directions to accelerate classroom use in three categories: (1) Focusing on teachers' professional learning and well-being, (2) Building a community, and (3) Taking AI-powered decisions.

Focusing on teachers' professional learning and well-being: I conducted regular meetings with teachers throughout the research and development processes. Although it was not the primary goal, all the project outcomes serve teachers' well-being by supporting their in-classroom flow. I developed Kart-ON to benefit from the increasing availability of smart mobile devices to promote active learning for group activities in a more engaging, accessible, and affordable way to a wider audience. Kart-ON demonstrated its efficiency in the qualitative and quantitative results, but I also observed that teachers did not use the Kart-ON to its full capacity. Only a few teachers promoted the use of creating self-made tangibles and introduced them as functional elements in the programming. Although they mentioned that it makes sense in the programming structure, their confidence in introducing this concept to their students was low. This is partly because their knowledge and experience in using AI are limited. One future direction is supporting the use and teaching process of intelligent applications by naturally explaining the key concepts about what makes the product intelligent.

Building a community: Although Kart-ON is a highly expressive programming framework as it utilizes *p5.js* code base, educational content created in part of the curriculum studies was limited to a few programming concepts. Transitioning from a research product to a widely used open-source product requires community support and the need to develop more educational content that can help teachers to ease their classroom flow. The successful programming platforms have a rich community of educators and students that develop and share unique educational content with other users [Resnick et al., 2009]. For example, the Scratch community has a rich set of resources that can help us develop more educational content. Kart-ON requires this community to become sustainable in terms of software development and educational content development.

Taking AI-powered decisions: In a classroom environment, not every moment can be engaging. The goal is generally to minimize the long disengaged moments to keep the students' motivation high. While using the dashboard, it is vital to help teachers to analyze ML-based reports by combining multiple data sources. Based on my experience, I would advise teachers to use ML-based engagement analysis when they already have some insights about students' characteristics, such as habits, likes, and dislikes. The ML models can only point out the engaged and disengaged moments based on some facial unit movements, body positions, and voice duration. This kind of guidance can be useful when students work together in a crowded classroom where the teacher cannot watch every group's behavior. However, in this scenario, the teacher still needs to take action, which can only be possible by re-evaluating the group dynamics and students' long-term behaviors.

9.3 Future Directions for the Dataset Considering the Limitations

My efforts in collecting real-life-like classroom engagement data was an initial step towards developing systems for automatic analysis of engagement. The video dataset is contributed to ML and Education research fields by presenting multimodal data with self-evaluation scores. It is the first dataset that presents group activity engagement in a multimodal format with student self-evaluation scores. However, as I elaborated in Section 6.1.5, the dataset is limited in terms of the number of samples and representation variety. This limited representation of the affective, cultural, behavioral, cognitive, and pedagogical require the careful attention of future researchers to investigate the representability and bias of their model.Before increasing the number of samples, an interdisciplinary collaboration of researchers from computer science, psychology, and education is much needed to determine the requirements for a set of reliable metrics for classroom engagement that can help researchers to build models that can support different learning styles varieties and reduce the racial and cultural bias.

The baseline architectures: Existing computational models on engagement recognition rely little on cognitive theories and still lack the required implementation of cognitive models while considering a large variety of interactions [Baveye et al., 2018]. Furthermore, they present a limited representation of group activities. As a result, I developed the DL models upon the state-of-the-art models and architectures that proved their effectiveness in video action recognition tasks. Our models can be used as a starting point for researchers to develop more sophisticated models. However, the baseline architecture is limited in terms of the number of layers and the number of parameters. Our experiments demonstrated that carefully selecting pre-trained feature extractors in a transfer learning approach is the most important factor in achieving good accuracy. Currently, the most effective and high-accuracy models are run on top of MobileNet (for face-area-fed models) and MoViNet (group-activity-video-fed models) architectures. I encourage researchers to test different feature extractors and mid-level representations that could increase model accuracy significantly and experiment with developing more sophisticated architectures to develop more sophisticated models.

Interpreting the data and model to middle school domain: The dataset is collected in a university setting where participants are from the same faculty. As I elaborated in more detail in Section 6.1.2, collecting data from university students has several advantages, such as easier ethical approval and better self-evaluation. Interpreting the results of studies conducted with university students for middle school students can be challenging, as developmental, cognitive, and affective abilities between these two groups show significant differences. Based on my observations and experiences in working with both groups, I suggest following future paths that researchers can take to increase the generalizability of their findings to the middle school domain:

1. The classroom engagement checklist was adapted from Wang et al.'s Classroom Inventory, which was already covering middle school use cases. But, future researchers can tweak the measures and assessments that are appropriate for the developmental stage and abilities of middle school students to use in the self-evaluation or independent observation processes.

2. I suggest researchers carefully consider the context in which the technology is being used. The learning environment in middle school may be different from that of a university, and students may have different expectations and motivations for learning. Researchers can account for these differences by conducting studies in middle school classrooms or by collecting data from middle school students and teachers.

3. The field needs transparent practices such as listing the limitations and being transparent about the extent to which their findings can be generalized. Similar to the open-source distribution of the *Classroom Engagement Dataset*, the anonymized distribution of the future sets without the video recordings can help accelerate the research in the domain.

4. In the data collection process, we balanced the dataset by selecting a set of activities that students find interesting but, at the same time, a bit boring. In the middle school setting, we cannot follow this type of intervention to support students' well-being. As researchers, we have the responsibility to maintain or improve the students' well-being when we involve students in our user studies. So, to balance the dataset, future researchers can either collect more data and regularly conduct exploratory analysis or develop synthetic data generation techniques.

5. Researchers can use transfer learning to adapt a pre-trained model developed using university student data to the middle school domain. Transfer learning is a technique that involves reusing a pre-trained model for a new task. In this case, this can be achieved by fine-tuning the model using the middle school data. Furthermore, foundation models have become a standard development path in the natural language processing (NLP) field. The computer vision field still has limited models that show similar success in the NLP field. The advances in foundation models in computer vision and multimodal data, can support researchers with limited domain-specific knowledge to generate highly-capable models.

9.4 Ethical Considerations

Throughout my Ph.D. studies, I have been working on developing systems that can help teachers and students to improve their learning experience. Working together with and for children requires careful consideration of the ethical implications of each step. In this section, I will discuss the ethical considerations of my research and suggest some guidelines for future researchers.

Our data pipeline and dashboard are designed to help students and teachers to observe the repeating patterns and help them to interpret their engagement levels. Yet, one can be concerned about using the dashboard as a classroom surveillance tool that can process personally-identifiable data that classifies behavior, attitudes, and preferences. Privacy concerns are long-lasting challenges of the learning analytics field [Lang et al., 2020, Slade et al., 2019, Arnold and Sclater, 2017, Prinsloo and Slade, 2017, Drachsler and Greller, 2016, Prinsloo and Slade, 2015]. Williamson et al. list four emerging challenges while developing LADs [Williamson and Kizilcec, 2022]: (1) Protecting participants' privacy while also including enough demographic information, (2) Surveillance concerns, (3) Neglecting pedagogies that fall outside of the dominant narrative and (4) Making LADs maintainable in terms of software development. The discussion below, categorized by these emerging issues, summarizes suggestions for adapters of the prediction model and dashboard.

1. **Participant Identities:** Our dashboard and model do not require collecting any demographic information. If a third party adapts the model and dashboard, they can run the models anonymously without requiring any identifiers. In the dashboard usage, teachers have access to all data, but students do not have access to other students' engagement levels. Currently, I identified two main users: Teachers and Students. Demographic information can be useful for policymakers when they make nationwide decisions. At this point, the engagement information should be aggregated to protect personal identifiers.

2. **Surveillance Concerns:** Teachers have access to personally identifiable data of their students. A student can only see video data from other members of the study group. Our deep learning architectures do not submit any information to third-party software. Our dashboard currently runs on Observable, but the Observable platform does not store any data when the notebook is on run-time. In the dashboard design, I aimed to achieve minimum surveillance concerns. I included a step-by-step explanation of data usage in an end-user-readable way, and I also suggested this approach to adapters of the system.

3. **Implicit Pedagogies:** By using the dashboard, students and teachers can explore their learning patterns that fall behind the dominant narrative. The dashboard aims to help students only interpret their engagement levels. If the adapters intend to give suggestions based on their pedagogic approach, they should carefully support their arguments by showing direct links to ML model features.

4. **Software Maintainability:** Presenting the dashboard on Observable improves the software maintainability significantly. Using Observable, researchers and developers can fork the interactive notebook and create custom dashboards based on their needs. They can also access the data pre-processing scripts and DL model training codes through the project's GitHub repo[4], which is currently active and open-source. I shared these resources with a share-a-like license, so adapters should also make their code open-source to increase maintainability.

.

[4] `https://github.com/asabuncuoglu13/classroom-engagement-dataset`

Chapter 10

CONCLUSION

This dissertation contributes to the field of human-computer interaction (HCI) by presenting a set of tools, activities and systems to introduce programming education with a focus on promoting classroom engagement in an affordable way (outlined below).

(I) In the first step of my Ph.D., I developed an *intelligent mobile application that can augment the capabilities of a paper programming setup* (Kart-ON). The development process followed an iterative flow where I met with students in individual, group and classroom settings. I regularly met with information and communication technology (ICT) teachers and middle school students.

(II) I created a *custom programming curriculum accompanying the paper programming setup.* The curriculum was *applied in three middle schools by the ICT teachers* for one semester. The learnings from the in-classroom interactions, and the curriculum study results, I updated the programming environment with the curated design considerations.

(III) I developed *a system that analyzes student engagement in a real-world classroom environment.* I trained a set of face and video-based deep learning models using the eight-hour-long recordings that I collected from group learning activities (Ntotal=34) . The dataset has become an initial step towards developing a multimodal model that can effectively predict classroom engagement using a single camera in an affordable fashion while considering the multi-component definition of engagement.

10.1 Revisiting Contributions

This dissertation presented my doctoral research on developing applications and deep learning models for building classroom engagement in K-12 schools. Throughout my research, I focused on establishing open-source, reliable intelligent systems for (1) Designing affordable programming environments using the advantage of mobile devices to enhance collaboration and engagement in crowded classrooms. (2) Providing end-to-end classroom solutions for programming education to develop computational thinking while maintaining engagement, (3) Analyzing engagement automatically in a multimodal fashion using already-available mobile device cameras in the classroom. These three main research goals led me to develop multiple tangible research outputs that I deployed open-source:

- **Kart-ON: An Affordable and Engaging Programming Platform:** The platform is open-source and showed its efficiency in both usability studies and long-term curriculum studies. The intelligent mobile application, helper web tools for teachers, and all related material is open-source at my Github Repository under *Kart-ON Project Organisation*.

- **Updated Curricula for Paper Programming:** Building upon the existing Turkish national curriculum, the Kart-ON curriculum includes 12-week of open-access, ready-to-apply educational materials to allow educators to easily adopt paper-programming tangible activities in their classrooms. Programming activities are designed to increase the affordability and promote group activities. Additionally, I published independent two-hour-long workshop materials to introduce programming using Kart-ON.

- **An Audio-Visual Engagement Dataset:** The dataset is the initial step towards achieving a classroom engagement prediction system that considers different learning styles and active classroom environment. The public dataset can be easily downloaded using the Download Script and used for further research.

- **Deep Learning Models:** The MoViNet-based models can run on any modern, everyday laptop, which increases the affordability and adoptability. The best model achieved 74% accuracy on my test-cases, which proves itself as a reliable starting point for future research.

- **Interactive Dashboard:** The modular and easy-to-update notebook style engagement prediction dashboard help students interpret their engagement levels and keep the process student-centric. The dashboard is developed interactively with students on Observable platform, and is still available at Observable Notebook.

10.2 Broader Implications

This dissertation presented novel intelligent systems to classroom engagement by combining the affordances of mobile devices and deep learning models. The research outcomes of this dissertation can be used to improve the quality of education in K-12 schools in an affordable fashion. Our programming platform, Kart-ON, can be used in both socio-economically disadvantaged schools to increase the access, and socio-economically advantaged schools to promote engaging group activities. The updated curricula draw a path for teachers to adopt this new approach with example activities. The engagement dataset and deep learning models can be used to develop more reliable engagement prediction systems using a few camera integration. The interactive dashboard promotes a student-centric approach to interpret classroom engagement levels.

From a broder perspective, this dissertation is only a small step towards the building equity in classrooms. Recent advances in computing technologies can be used to improve the quality, accessibility and affordability of education in K-12 schools. The affordances of mobile devices and internet infrastructure can keep the students connected to the state-of-the-art technologies. Recent deep learning models can be used to develop more reliable, affordable and natural intelligent systems for classroom engagement. The research outcomes of this dissertation can be used as a starting point for future research to improve the quality of education in K-12 schools.

10.3 Concluding Remarks

Building, analysing and predicting classroom engagement is a complex task. The research outcomes of this dissertation are only a small step towards the building equity in classrooms. I published all the material open-source to allow other researchers to build upon my work. I hope that my work will be useful for the education community and will help to improve the quality of education.

BIBLIOGRAPHY

[Acar et al., 2017] Acar, E., Hopfgartner, F., and Albayrak, S. (2017). A comprehensive study on mid-level representation and ensemble learning for emotional analysis of video material. *Multimedia Tools and Applications*, 76(9):11809–11837.

[Amershi et al., 2019] Amershi, S., Weld, D., Vorvoreanu, M., Fourney, A., Nushi, B., Collisson, P., Suh, J., Iqbal, S., Bennett, P. N., Inkpen, K., Teevan, J., Kikin-Gil, R., and Horvitz, E. (2019). Guidelines for Human-AI Interaction. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, pages 1–13, Glasgow Scotland Uk. ACM.

[Andrade, 2019] Andrade, H. L. (2019). A critical review of research on student self-assessment. *Frontiers in Education*, 4.

[Andriluka et al., 2014] Andriluka, M., Pishchulin, L., Gehler, P., and Schiele, B. (2014). 2d human pose estimation: New benchmark and state of the art analysis. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 3686–3693.

[Antle and Hourcade, 2022] Antle, A. N. and Hourcade, J. P. (2022). Research in Child–Computer Interaction: Provocations and envisioning future directions. *International Journal of Child-Computer Interaction*, 32:100374.

[Arduino, 2021] Arduino (2021). Arduino - Home.

[Arnold and Sclater, 2017] Arnold, K. E. and Sclater, N. (2017). Student perceptions of their privacy in leaning analytics applications. In *Proceedings of the Seventh International Learning Analytics & Knowledge Conference*, pages 66–69, Vancouver British Columbia Canada. ACM.

[Ball et al., 2016] Ball, T., Protzenko, J., Bishop, J., Moskal, M., de Halleux, J., Braun, M., Hodges, S., and Riley, C. (2016). Microsoft touch develop and the bbc micro:bit. In *2016 IEEE/ACM 38th International Conference on Software Engineering Companion (ICSE-C)*, pages 637–640.

[Baltrusaitis et al., 2015] Baltrusaitis, T., Mahmoud, M., and Robinson, P. (2015). Cross-dataset learning and person-specific normalisation for automatic action unit detection. In *2015 11th IEEE International Conference and Workshops on Automatic Face and Gesture Recognition (FG)*, pages 1–6. IEEE.

[Baveye et al., 2018] Baveye, Y., Chamaret, C., Dellandrea, E., and Chen, L. (2018). Affective Video Content Analysis: A Multidisciplinary Insight. *IEEE Transactions on Affective Computing*, 9(4):396–409.

[Bell and Vahrenhold, 2018] Bell, T. and Vahrenhold, J. (2018). *CS Unplugged—How Is It Used, and Does It Work?*, pages 497–521. Springer International Publishing, Cham.

[Bennett, 2018] Bennett, L. (2018). Students' learning responses to receiving dashboard data: Research report. Technical report.

[Black and Wiliam, 1998] Black, P. and Wiliam, D. (1998). Assessment and Classroom Learning. *Assessment in Education: Principles, Policy & Practice*, 5(1):7–74. Publisher: Routledge _eprint: https://doi.org/10.1080/0969595980050102.

[Bond et al., 2020] Bond, M., Buntins, K., Bedenlier, S., Zawacki-Richter, O., and Kerres, M. (2020). Mapping research in student engagement and educational technology in higher education: a systematic evidence map. *International Journal of Educational Technology in Higher Education*, 17(1):2.

[Bosch et al., 2016] Bosch, N., D'Mello, S. K., Baker, R. S., Ocumpaugh, J., Shute, V., Ventura, M., Wang, L., and Zhao, W. (2016). Detecting student emotions in computer-enabled classrooms. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*, IJCAI'16, page 4125–4129. AAAI Press.

[Braun and Clarke, 2019] Braun, V. and Clarke, V. (2019). Reflecting on reflexive thematic analysis. *Qualitative Research in Sport, Exercise and Health*, 11(4):589–597.

[Cao et al., 2019] Cao, Z., Hidalgo, G., Simon, T., Wei, S.-E., and Sheikh, Y. (2019). OpenPose: Realtime multi-person 2d pose estimation using part affinity fields.

[Carreira and Zisserman, 2018] Carreira, J. and Zisserman, A. (2018). Quo vadis, action recognition? a new model and the kinetics dataset.

[Dalal and Triggs, 2005] Dalal, N. and Triggs, B. (2005). Histograms of oriented gradients for human detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 886–893.

[Delgado et al., 2021] Delgado, K., Origgi, J. M., Hasanpoor, T., Yu, H., Allessio, D., Arroyo, I., Lee, W., Betke, M., Woolf, B., and Bargal, S. A. (2021). Student engagement dataset. In *2021 IEEE/CVF International Conference on Computer Vision Workshops (ICCVW)*, pages 3621–3629. IEEE.

[Deng et al., 2009] Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. (2009). Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee.

[Deng et al., 2019] Deng, X., Wang, D., Jin, Q., and Sun, F. (2019). Arcat: A tangible programming tool for dfs algorithm teaching. In *Proceedings of the 18th ACM International Conference on Interaction Design and Children*, IDC '19, page 533–537, New York, NY, USA. Association for Computing Machinery.

[Denner et al., 2014] Denner, J., Werner, L., Campe, S., and Ortiz, E. (2014). Pair programming: Under what conditions is it advantageous for middle school students? *Journal of Research on Technology in Education*, 46(3):277–296.

[Dewan et al., 2019] Dewan, M. A. A., Murshed, M., and Lin, F. (2019). Engagement detection in online learning: a review. *Smart Learning Environments*, page 21.

[Dhar et al., 2020] Dhar, S., Guo, J., Liu, J., Tripathi, S., Kurup, U., and Shah, M. (2020). On-Device Machine Learning: An Algorithms and Learning Theory Perspective. arXiv:1911.00623 [cs, stat].

[Dillenbourg, 1999] Dillenbourg, P. (1999). What do you mean by collaborative learning?

[Drachsler and Greller, 2016] Drachsler, H. and Greller, W. (2016). Privacy and analytics: it's a DELICATE issue a checklist for trusted learning analytics. In *Proceedings of the Sixth International Conference on Learning Analytics & Knowledge - LAK '16*, pages 89–98, Edinburgh, United Kingdom. ACM Press.

[Druin, 2002a] Druin, A. (2002a). The role of children in the design of new technology. *Behaviour & Information Technology*, 21(1):1–25.

[Druin, 2002b] Druin, A. (2002b). The role of children in the design of new technology. *Behaviour & Information Technology*, 21(1):1–25.

[Fang et al., 2018] Fang, H.-S., Xie, S., Tai, Y.-W., and Lu, C. (2018). RMPE: Regional multi-person pose estimation. version: 5.

[Fernaeus and Tholander, 2006] Fernaeus, Y. and Tholander, J. (2006). Finding design qualities in a tangible programming space. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '06, page 447–456, New York, NY, USA. Association for Computing Machinery.

[Fernandez Nieto et al., 2022] Fernandez Nieto, G. M., Kitto, K., Buckingham Shum, S., and Martinez-Maldonado, R. (2022). Beyond the learning analytics dashboard: Alternative ways to communicate student data insights combining visualisation, narrative and storytelling. In *LAK22: 12th International Learning Analytics and Knowledge Conference*, pages 219–229. ACM.

[Fredricks et al., 2004] Fredricks, J. A., Blumenfeld, P. C., and Paris, A. H. (2004). School engagement: Potential of the concept, state of the evidence. *Review of Educational Research*, 74(1):59–109.

[Fu et al., 2009] Fu, F.-L., Su, R.-C., and Yu, S.-C. (2009). Egameflow: A scale to measure learners' enjoyment of e-learning games. *Computers & Education*, 52(1):101–112.

[Fuste and Schmandt, 2019] Fuste, A. and Schmandt, C. (2019). Hypercubes: A playful introduction to computational thinking in augmented reality. In *Extended Abstracts of the Annual Symposium on Computer-Human Interaction in Play Companion Extended Abstracts*, CHI PLAY '19 Extended Abstracts, page 379–387, New York, NY, USA. Association for Computing Machinery.

[Fyfe et al., 2014] Fyfe, E. R., McNeil, N. M., Son, J. Y., and Goldstone, R. L. (2014). Concreteness Fading in Mathematics and Science Instruction: a Systematic Review. *Educational Psychology Review*, 26(1):9–25.

[Gao et al., 2021] Gao, N., Saiedur Rahaman, M., Shao, W., and Salim, F. D. (2021). Investigating the Reliability of Self-Report Data in the Wild: The Quest for Ground Truth. In *Adjunct Proceedings of the 2021 ACM International Joint Conference on*

Pervasive and Ubiquitous Computing and Proceedings of the 2021 ACM International Symposium on Wearable Computers, UbiComp '21, pages 237–242, New York, NY, USA. Association for Computing Machinery. event-place: Virtual, USA.

[Girard, 2014] Girard, J. M. (2014). CARMA: Software for continuous affect rating and media annotation. *Journal of Open Research Software*, 2(1):e5.

[Goodfellow et al., 2013] Goodfellow, I., Warde-Farley, D., Mirza, M., Courville, A., and Bengio, Y. (2013). Maxout networks. In Dasgupta, S. and McAllester, D., editors, *Proceedings of the 30th International Conference on Machine Learning*, volume 28 of *Proceedings of Machine Learning Research*, pages 1319–1327, Atlanta, Georgia, USA. PMLR.

[Google MLKit, 2021] Google MLKit (2021). ML Kit — Google Developers.

[Goyal et al., 2016] Goyal, S., Vijay, R. S., Monga, C., and Kalita, P. (2016). Code bits: An inexpensive tangible computational thinking toolkit for k-12 curriculum. In *Proceedings of the TEI '16: Tenth International Conference on Tangible, Embedded, and Embodied Interaction*, TEI '16, page 441–447, New York, NY, USA. Association for Computing Machinery.

[Grover and Pea, 2013] Grover, S. and Pea, R. (2013). Computational thinking in k–12: A review of the state of the field. *Educational researcher*, 42(1):38–43.

[Grover et al., 2015] Grover, S., Pea, R., and Cooper, S. (2015). Designing for deeper learning in a blended computer science course for middle school students. *Computer Science Education*, 25:199–237.

[Gupta et al., 2022] Gupta, A., D'Cunha, A., Awasthi, K., and Balasubramanian, V. (2022). DAiSEE: Towards user engagement recognition in the wild.

[Hagras, 2018] Hagras, H. (2018). Toward human-understandable, explainable AI. *Computer*, 51(9):28–36.

[Hattie and Timperley, 2007] Hattie, J. and Timperley, H. (2007). The power of feedback. *Review of educational research*, 77(1):81–112.

[Hirsh-Pasek et al., 2015] Hirsh-Pasek, K., Zosh, J. M., Golinkoff, R. M., Gray, J. H., Robb, M. B., and Kaufman, J. (2015). Putting Education in "Educational" Apps. *Psychological Science in the Public Interest*, 16(1):3–34.

[Horn and Jacob, 2007a] Horn, M. S. and Jacob, R. J. (2007a). Tangible programming in the classroom with tern. In *CHI'07 extended abstracts on Human factors in computing systems*, pages 1965–1970.

[Horn and Jacob, 2007b] Horn, M. S. and Jacob, R. J. K. (2007b). Designing tangible programming languages for classroom use. In *Proceedings of the 1st International Conference on Tangible and Embedded Interaction*, TEI '07, page 159–162, New York, NY, USA. Association for Computing Machinery.

[Horn and Jacob, 2007c] Horn, M. S. and Jacob, R. J. K. (2007c). Designing tangible programming languages for classroom use. In *International Conference on Tangible, Embedded, and Embodied Interaction.*

[Horn et al., 2008] Horn, M. S., Solovey, E. T., and Jacob, R. J. K. (2008). Tangible Programming and Informal Science Learning: Making TUIs Work for Museums. In *Proceedings of the 7th International Conference on Interaction Design and Children*, IDC '08, pages 194–201, New York, NY, USA. Association for Computing Machinery. event-place: Chicago, Illinois.

[Hu et al., 2015] Hu, F., Zekelman, A., Horn, M., and Judd, F. (2015). Strawbies: Explorations in tangible programming. In *Proceedings of the 14th International Conference on Interaction Design and Children*, IDC '15, page 410–413, New York, NY, USA. Association for Computing Machinery.

[Hubwieser et al., 2015] Hubwieser, P., Giannakos, M. N., Berges, M., Brinda, T., Diethelm, I., Magenheim, J., Pal, Y., Jackova, J., and Jasute, E. (2015). A global snapshot of computer science education in k-12 schools. In *Proceedings of the 2015 ITiCSE on working group reports*, pages 65–83.

[Im and Rogers, 2021] Im, H. and Rogers, C. (2021). *Draw2Code: Low-Cost Tangible Programming for Creating AR Animations*, page 427–432. Association for Computing Machinery, New York, NY, USA.

[Iooss and Saltelli, 2017] Iooss, B. and Saltelli, A. (2017). *Introduction to Sensitivity Analysis*, pages 1103–1122. Springer International Publishing.

[Ishii and Ullmer, 1997] Ishii, H. and Ullmer, B. (1997). Tangible bits: towards seamless interfaces between people, bits and atoms. In *Proceedings of the ACM SIGCHI Conference on Human factors in computing systems*, pages 234–241.

[Jacobs, 2013] Jacobs, I. M. (2013). Modernizing Education and Preparing Tomorrow's Workforce through Mobile Technology. *World Economic Forum Global Information Technology Report*.

[Jolliffe and Cadima, 2016] Jolliffe, I. T. and Cadima, J. (2016). Principal component analysis: a review and recent developments. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 374(2065):20150202.

[Kafai and Burke, 2014] Kafai, Y. B. and Burke, Q. (2014). *Connected code: Why children need to learn programming*. Mit Press.

[Kahou et al., 2013] Kahou, S. E., Pal, C., Bouthillier, X., Froumenty, P., Gülçehre, c., Memisevic, R., Vincent, P., Courville, A., Bengio, Y., Ferrari, R. C., Mirza, M., Jean, S., Carrier, P.-L., Dauphin, Y., Boulanger-Lewandowski, N., Aggarwal, A., Zumer, J., Lamblin, P., Raymond, J.-P., Desjardins, G., Pascanu, R., Warde-Farley, D., Torabi, A., Sharma, A., Bengio, E., Côté, M., Konda, K. R., and Wu, Z. (2013). Combining modality specific deep neural networks for emotion recognition in video. In *Proceedings of the 15th ACM on International Conference on Multimodal Interaction*, ICMI '13, page 543–550, New York, NY, USA. Association for Computing Machinery.

[Kairouz et al., 2021] Kairouz, P., McMahan, H. B., Avent, B., Bellet, A., Bennis, M., Nitin Bhagoji, A., Bonawitz, K., Charles, Z., Cormode, G., Cummings, R., D'Oliveira, R. G. L., Eichner, H., El Rouayheb, S., Evans, D., Gardner, J., Garrett, Z., Gascón, A., Ghazi, B., Gibbons, P. B., Gruteser, M., Harchaoui, Z., He, C., He, L., Huo, Z., Hutchinson, B., Hsu, J., Jaggi, M., Javidi, T., Joshi, G., Khodak, M., Konecný, J., Korolova, A., Koushanfar, F., Koyejo, S., Lepoint, T., Liu, Y., Mittal, P., Mohri, M., Nock, R., Özgür, A., Pagh, R., Qi, H., Ramage, D., Raskar, R., Raykova, M., Song, D., Song, W., Stich, S. U., Sun, Z., Suresh, A. T., Tramèr, F., Vepakomma, P., Wang, J., Xiong, L., Xu, Z., Yang, Q., Yu, F. X., Yu, H., and Zhao, S. (2021). Advances and Open Problems in Federated Learning. *Foundations and Trends® in Machine Learning*, 14(1–2):1–210.

[Kondratyuk et al., 2021] Kondratyuk, D., Yuan, L., Li, Y., Zhang, L., Tan, M., Brown, M., and Gong, B. (2021). MoViNets: Mobile Video Networks for Efficient Video Recognition. arXiv:2103.11511 [cs].

[Kraemer et al., 2009] Kraemer, K. L., Dedrick, J., and Sharma, P. (2009). One laptop per child: Vision vs. reality. *Commun. ACM*, 52(6):66–73.

[Kuh, 2008] Kuh, G. D. (2008). High-impact educational practices. *Peer Review*, 10(4):30–31.

[Lang et al., 2020] Lang, C., Woo, C., and Sinclair, J. (2020). Quantifying data sensitivity: precise demonstration of care when building student prediction models. In *Proceedings of the Tenth International Conference on Learning Analytics & Knowledge*, pages 655–664, Frankfurt Germany. ACM.

[Laura Musikanski et al., 2018] Laura Musikanski, Havens, J. C., and Gunsch, G. H. (2018). IEEE P7010 Well-being Metrics Standard for Autonomous and Intelligent Systems. Technical report, IEEE.

[Lavin et al., 2022] Lavin, A., Gilligan-Lee, C. M., Visnjic, A., Ganju, S., Newman, D., Ganguly, S., Lange, D., Baydin, A. G., Sharma, A., Gibson, A., Zheng, S., Xing, E. P., Mattmann, C., Parr, J., and Gal, Y. (2022). Technology readiness levels for machine learning systems. *Nature Communications*, 13(1):6039.

[LEGO, 2021] LEGO (2021). LEGO® MINDSTORMS® About — Official LEGO® Shop GB.

[Lu and Meeks, 2022] Lu, S. and Meeks, E. (2022). Viz palette.

[Luo et al., 2022] Luo, C., Song, S., Xie, W., Shen, L., and Gunes, H. (2022). Learning multi-dimensional edge feature-based AU relation graph for facial action unit recognition. In *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence*, pages 1239–1246.

[Malandrakis et al., 2011] Malandrakis, N., Potamianos, A., Evangelopoulos, G., and Zlatintsi, A. (2011). A supervised approach to movie emotion tracking. In *2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2376–2379.

[Maloney et al., 2010] Maloney, J., Resnick, M., Rusk, N., Silverman, B., and Eastmond, E. (2010). The scratch programming language and environment. *ACM Transactions on Computing Education (TOCE)*, 10(4):1–15.

[Marshall, 2007] Marshall, P. (2007). Do tangible interfaces enhance learning? In *Proceedings of the 1st international conference on Tangible and embedded interaction*, pages 163–170.

[Martinez-Maldonado et al., 2015] Martinez-Maldonado, R., Pardo, A., Mirriahi, N., Yacef, K., Kay, J., and Clayphan, A. (2015). The LATUX workflow: designing and deploying awareness tools in technology-enabled learning settings. In *Proceedings of the Fifth International Conference on Learning Analytics And Knowledge*, pages 1–10. ACM.

[Marzano, 2007] Marzano, R. J. (2007). *The art and science of teaching: A comprehensive framework for effective instruction.* Ascd.

[Maulsby et al., 1993] Maulsby, D., Greenberg, S., and Mander, R. (1993). Prototyping an intelligent agent through wizard of oz. In *Proceedings of the INTERACT'93 and CHI'93 conference on Human factors in computing systems*, pages 277–284.

[MBlock, 2021] MBlock (2021). Program Neuron with mBlock — mBlock.

[McInnes et al., 2018] McInnes, L., Healy, J., and Melville, J. (2018). Umap: Uniform manifold approximation and projection for dimension reduction. *arXiv preprint arXiv:1802.03426*.

[MEB, 2018] MEB (2018). Teaching guide - information and software technologies (5th and 6th grades).

[Molnar, 2022] Molnar, C. (2022). *Interpretable ML Book.*

[Moors et al., 2018] Moors, L., Luxton-Reilly, A., and Denny, P. (2018). Transitioning from Block-Based to Text-Based Programming Languages. In *2018 International Conference on Learning and Teaching in Computing and Engineering (LaTICE)*, pages 57–64. IEEE.

[Muldner et al., 2015] Muldner, K., Wixon, M., Rai, D., Burleson, W., Woolf, B., and Arroyo, I. (2015). Exploring the Impact of a Learning Dashboard on Student Affect. pages 307–317, Cham. Springer International Publishing.

[Nicolaou et al., 2011] Nicolaou, M. A., Gunes, H., and Pantic, M. (2011). Continuous prediction of spontaneous affect from multiple cues and modalities in valence-arousal space. *IEEE Transactions on Affective Computing*, 2(2):92–105.

[Nie et al., 2017] Nie, X., Feng, J., Xing, J., and Yan, S. (2017). Generative partition networks for multi-person pose estimation. version: 2.

[Nie et al., 2019] Nie, X., Zhang, J., Yan, S., and Feng, J. (2019). Single-stage multi-person pose machines. version: 1.

[Nori et al., 2019] Nori, H., Jenkins, S., Koch, P., and Caruana, R. (2019). InterpretML: A unified framework for machine learning interpretability.

[Observable, 2023] Observable (2023). Observable Notebooks.

[OECD, 2020] OECD (2020). *PISA 2018 Results (Volume V)*.

[OECD, 2022] OECD (2022). *Education at a Glance 2022: OECD Indicators*. Education at a Glance. OECD.

[O'Brien and Toms, 2013] O'Brien, H. L. and Toms, E. G. (2013). Examining the generalizability of the user engagement scale (ues) in exploratory search. *Information Processing & Management*, 49(5):1092–1107.

[Paleyes et al., 2023] Paleyes, A., Urma, R.-G., and Lawrence, N. D. (2023). Challenges in Deploying Machine Learning: a Survey of Case Studies. *ACM Computing Surveys*, 55(6):1–29. arXiv:2011.09926 [cs].

[Papert, 1999] Papert, S. (1999). What is logo? who needs it. *Logo philosophy and implementation*, pages 4–16.

[Papineni et al., 2002] Papineni, K., Roukos, S., Ward, T., and Zhu, W.-J. (2002). Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318.

[Pareja Roblin et al., 2014] Pareja Roblin, N. N., Ormel, B. J., McKenney, S. E., Voogt, J. M., and Pieters, J. M. (2014). Linking research and practice through teacher communities: A place where formal and practical knowledge meet? *European Journal of Teacher Education*, 37(2):183–203.

[Parsons and Haden, 2007] Parsons, D. and Haden, P. (2007). Programming osmosis: Knowledge transfer from imperative to visual programming environments.

[Pattis et al., 1981] Pattis, R., Roberts, J., and Stehlik, M. (1981). Karel the robot. *A gentele introduction to the Art of Programming.*

[Pozdniakov et al., 2022] Pozdniakov, S., Martinez-Maldonado, R., Tsai, Y.-S., Cukurova, M., Bartindale, T., Chen, P., Marshall, H., Richardson, D., and Gasevic, D. (2022). The question-driven dashboard: How can we design analytics interfaces aligned to teachers' inquiry? In *LAK22: 12th International Learning Analytics and Knowledge Conference*, pages 175–185. ACM.

[Prinsloo and Slade, 2015] Prinsloo, P. and Slade, S. (2015). Student privacy self-management: implications for learning analytics. In *Proceedings of the Fifth International Conference on Learning Analytics And Knowledge*, pages 83–92, Poughkeepsie New York. ACM.

[Prinsloo and Slade, 2017] Prinsloo, P. and Slade, S. (2017). An elephant in the learning analytics room: the obligation to act. In *Proceedings of the Seventh International Learning Analytics & Knowledge Conference*, pages 46–55, Vancouver British Columbia Canada. ACM.

[Python Docs, 2021] Python Docs (2021). turtle — Turtle graphics — Python 3.9.2 documentation.

[Radford et al., 2022] Radford, A., Kim, J. W., Xu, T., Brockman, G., McLeavey, C., and Sutskever, I. (2022). Robust speech recognition via large-scale weak supervision. *arXiv preprint arXiv:2212.04356.*

[Raffle et al., 2004] Raffle, H. S., Parkes, A. J., and Ishii, H. (2004). Topobo: a constructive assembly system with kinetic memory. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 647–654.

[Reas and Fry, 2007] Reas, C. and Fry, B. (2007). *Processing: a programming handbook for visual designers and artists*. Mit Press.

[Resnick et al., 2009] Resnick, M., Maloney, J., Monroy-Hernández, A., Rusk, N., Eastmond, E., Brennan, K., Millner, A., Rosenbaum, E., Silver, J., Silverman, B., and Kafai, Y. (2009). Scratch: Programming for all. *Commun. ACM*, 52(11):60–67.

[Resnick and Silverman, 2005] Resnick, M. and Silverman, B. (2005). Some reflections on designing construction kits for kids. In *Proceedings of the 2005 conference on Interaction design and children*, pages 117–122.

[Ribeiro et al., 2016] Ribeiro, M. T., Singh, S., and Guestrin, C. (2016). "why should i trust you?": Explaining the predictions of any classifier.

[Sabuncuoglu, 2020] Sabuncuoglu, A. (2020). Tangible music programming blocks for visually impaired children. In *Proceedings of the Fourteenth International Conference on Tangible, Embedded, and Embodied Interaction*, pages 423–429.

[Sabuncuoglu et al., 2023a] Sabuncuoglu, A., Besevli, C., and Sezgin, T. M. (2023a). Towards building child-centered machine learning pipelines: Use cases from k-12 and higher-education.

[Sabuncuoglu et al., 2023b] Sabuncuoglu, A., Besevli, C., and Sezgin, T. M. (2023b). Towards building child-centered machine learning pipelines: Use cases from k-12 and higher-education. *arXiv preprint arXiv:2304.09532*.

[Sabuncuoğlu et al., 2018] Sabuncuoğlu, A., Erkaya, M., Buruk, O. T., and Göksun, T. (2018). Code notes: designing a low-cost tangible coding tool for/with children. In *17th ACM Conference on Interaction Design and Children (IDC '18)*, pages 644–649.

[Sabuncuoğlu and Sezgin, 2020] Sabuncuoğlu, A. and Sezgin, M. (2020). Kart-on: affordable early programming education with shared smartphones and easy-to-find materials. In *Proceedings of the 25th International Conference on Intelligent User Interfaces Companion*, pages 116–117.

[Sabuncuoglu and Sezgin, 2022a] Sabuncuoglu, A. and Sezgin, T. M. (2022a). Exploring children's use of self-made tangibles in programming. *arXiv preprint arXiv:2210.06258*.

[Sabuncuoglu and Sezgin, 2022b] Sabuncuoglu, A. and Sezgin, T. M. (2022b). Kart-on: An extensible paper programming strategy for affordable early programming education. *Proceedings of the ACM on Human-Computer Interaction*, 6(EICS):1–18.

[Sabuncuoglu and Sezgin, 2022c] Sabuncuoglu, A. and Sezgin, T. M. (2022c). Prototyping products using web-based ai tools: Designing a tangible programming environment with children. In *6th FabLearn Europe/MakeEd Conference 2022*, pages 1–6.

[Sabuncuoglu and Sezgin, 2023] Sabuncuoglu, A. and Sezgin, T. M. (2023). Developing a multimodal classroom engagement analysis dashboard for higher-education students. *Proceedings of the ACM on Human-Computer Interaction*, 7(EICS):1–23.

[Sabuncuoğlu et al., 2021] Sabuncuoğlu, A., Yantaç, A. E., and Sezgin, T. M. (2021). Teaching k-12 classrooms data programming: A three-week workshop with online and unplugged activities. In *VisActivities: 2nd IEEE VIS Workshop on Data Vis Activities to Facilitate Learning, Reflecting, Discussing, and Designing*.

[Saini and Goel, 2019] Saini, M. K. and Goel, N. (2019). How smart are smart classrooms? a review of smart classroom technologies. *ACM Comput. Surv.*, 52(6).

[Sclater et al., 2016] Sclater, N., Peasgood, A., and Mullan, J. (2016). Learning analytics in higher education: A review of UK and international practice. Technical report.

[Senchanka,] Senchanka, P. Example on-device model personalization with TensorFlow Lite — The TensorFlow Blog.

[Sentance et al., 2019] Sentance, S., Waite, J., and Kallia, M. (2019). Teaching computer programming with primm: a sociocultural perspective. *Computer Science Education*, 29(2-3):136–176.

[Shabaninejad et al., 2022] Shabaninejad, S., Khosravi, H., Abdi, S., Indulska, M., and Sadiq, S. (2022). Incorporating explainable learning analytics to assist educators with identifying students in need of attention. In *Proceedings of the Ninth ACM Conference on Learning @ Scale*, pages 384–388. ACM.

[Simonyan and Zisserman, 2014] Simonyan, K. and Zisserman, A. (2014). Two-stream convolutional networks for action recognition in videos.

[Slade et al., 2019] Slade, S., Prinsloo, P., and Khalil, M. (2019). Learning analytics at the intersections of student trust, disclosure and benefit. In *Proceedings of the 9th International Conference on Learning Analytics & Knowledge*, pages 235–244, Tempe AZ USA. ACM.

[So, 1964] So, I. (1964). Cognitive development in children: Piaget development and learning. *Journal, of Research in Science Teaching*, 2:176–186.

[Sofia da Silva Antunes, 2020] Sofia da Silva Antunes, R. (2020). DATUS: Dashboard assessment usability model: A case study with student dashboards. mastersthesis.

[Song et al., 2021] Song, L., Yu, G., Yuan, J., and Liu, Z. (2021). Human pose estimation and its application to action recognition: A survey. *Journal of Visual Communication and Image Representation*, 76:103055.

[Sümer et al., 2021] Sümer, O., Goldberg, P., D'Mello, S., Gerjets, P., Trautwein, U., and Kasneci, E. (2021). Multimodal engagement analysis from facial videos in the classroom.

[Tada and Tanaka, 2015] Tada, K. and Tanaka, J. (2015). Tangible programming environment using paper cards as command objects. *Procedia Manufacturing*, 3:5482–5489. 6th International Conference on Applied Human Factors and Ergonomics (AHFE 2015) and the Affiliated Conferences, AHFE 2015.

[Tan et al., 2018] Tan, C., Sun, F., Kong, T., Zhang, W., Yang, C., and Liu, C. (2018). A survey on deep transfer learning.

[Tensorflow, 2022] Tensorflow (2022). Embedding projector - visualization of high-dimensional data.

[Tian et al., 2001] Tian, Y.-l., Kanade, T., and Cohn, J. F. (2001). Recognizing action units for facial expression analysis. *IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE*, 23(2):19.

[Tillmann et al., 2012] Tillmann, N., Moskal, M., De Halleux, J., Fahndrich, M., and Burckhardt, S. (2012). Touchdevelop: app development on mobile devices. In *Proceedings* *of the ACM SIGSOFT 20th International Symposium on the Foundations of Software Engineering*, pages 1–2.

[Tissenbaum et al., 2021] Tissenbaum, M., Weintrop, D., Holbert, N., and Clegg, T. (2021). The case for alternative endpoints in computing education. *British Journal of Educational Technology*, 52(3):1164–1177.

[Tran et al., 2017] Tran, D., Wang, H., Torresani, L., Ray, J., LeCun, Y., and Paluri, M. (2017). A closer look at spatiotemporal convolutions for action recognition.

[Twin Science and Robotics, 2021] Twin Science and Robotics (2021). Twin Science — Science Kit - Robotics, Coding and Education Kits.

[UNESCO, 2015] UNESCO (2015). Rethinking education: Towards a global common good? Technical report.

[UNESCO, 2019] UNESCO (2019). Artificial intelligence in education, compendium of promising initiatives: Mobile Learning Week 2019. Technical report.

[UNESCO, 2021] UNESCO (2021). Policy Guidance on AI for Children. Technical report.

[UXCam, 2021] UXCam (2021). Deliver the perfect app experience.

[Van der Maaten and Hinton, 2008] Van der Maaten, L. and Hinton, G. (2008). Visualizing data using t-sne. *Journal of machine learning research*, 9(11).

[Van Mechelen et al., 2014] Van Mechelen, M., Gielen, M., vanden Abeele, V., Laenen, A., and Zaman, B. (2014). Exploring challenging group dynamics in participatory design with children. In *Proceedings of the 2014 Conference on Interaction Design and Children*, IDC '14, page 269–272, New York, NY, USA. Association for Computing Machinery.

[Wang and Schmid, 2013] Wang, H. and Schmid, C. (2013). Action recognition with improved trajectories. In *2013 IEEE International Conference on Computer Vision*, pages 3551–3558.

[Wang et al., 2014] Wang, Z., Bergin, C., and Bergin, D. A. (2014). Measuring engagement in fourth to twelfth grade classrooms: The classroom engagement inventory. 29(4):517–535.

[Whitehill et al., 2014] Whitehill, J., Serpell, Z., Lin, Y.-C., Foster, A., and Movellan, J. R. (2014). The faces of engagement: Automatic recognition of student engagement-from facial expressions. *IEEE TRANSACTIONS ON AFFECTIVE COMPUTING*, 5(1):13.

[Williamson and Kizilcec, 2022] Williamson, K. and Kizilcec, R. (2022). A Review of Learning Analytics Dashboard Research in Higher Education: Implications for Justice, Equity, Diversity, and Inclusion. In *LAK22: 12th International Learning Analytics and Knowledge Conference*, pages 260–270, Online USA. ACM.

[Wing, 2006] Wing, J. M. (2006). Computational thinking. *Communications of the ACM*, 49(3):33–35.

[Wu et al., 2018] Wu, X., He, R., Sun, Z., and Tan, T. (2018). A Light CNN for Deep Face Representation with Noisy Labels. arXiv:1511.02683 [cs].

[Yan et al., 2018] Yan, S., Xiong, Y., and Lin, D. (2018). Spatial temporal graph convolutional networks for skeleton-based action recognition. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

[Yu et al., 2021] Yu, H., Gupta, A., Lee, W., Arroyo, I., Betke, M., Allesio, D., Murray, T., Magee, J., and Woolf, B. P. (2021). Measuring and integrating facial expressions and head pose as indicators of engagement and affect in tutoring systems. In Sottilare, R. A. and Schwarz, J., editors, *Adaptive Instructional Systems. Adaptation Strategies and Methods*, volume 12793, pages 219–233. Springer International Publishing. Series Title: Lecture Notes in Computer Science.

[Yue-Hei Ng et al., 2015] Yue-Hei Ng, J., Hausknecht, M., Vijayanarasimhan, S., Vinyals, O., Monga, R., and Toderici, G. (2015). Beyond short snippets: Deep networks for video classification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4694–4702.

[Yun et al., 2020] Yun, W.-H., Lee, D., Park, C., Kim, J., and Kim, J. (2020). Automatic recognition of children engagement from facial video using convolutional neural networks. *IEEE Transactions on Affective Computing*, 11(4):696–707.

[Zhang et al., 2014] Zhang, X., Yin, L., Cohn, J. F., Canavan, S., Reale, M., Horowitz, A., Liu, P., and Girard, J. M. (2014). BP4d-spontaneous: a high-resolution spontaneous 3d dynamic facial expression database. *Image and Vision Computing*, 32(10):692–706.

[Zuckerman et al., 2005] Zuckerman, O., Arida, S., and Resnick, M. (2005). Extending tangible interfaces for education: digital montessori-inspired manipulatives. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 859–868.

Appendix A

LEARNING GOALS

The learning goals are adopted from Turkish National ICT Curriculum for 5th and 6th grades. [MEB, 2018].

G1. Fostering problem-solving concepts and approaches
G1.1. Proposes solutions to the problems faced in daily life.
G1.2. Solves a problem using the appropriate steps. Knows a problem can be solved via different solutions.
G1.3. Explains the problem types by using fundamental concepts and definitions.
G1.4. Realizes the steps to be followed during the problem-solving process.
G1.5. Analyzes a given problem.
G1.6. Explains the required variables, constants, and operators to solve the problem.
G1.7. Gives examples of operators that can be used in problem-solving.
G1.8. Gives examples of expressions and equations in problem-solving.
G1.9. Gives examples of operator precedence in problem-solving.
G1.10. Uses operators to solve a problem.
G1.11. Proposes solutions by using expressions and equations in a problem given.
G1.12. Explains the concept of algorithm.
G1.13 Develops algorithm for the solution of a problem.
G1.14 Explains the flow chart components and functions.
G1.15 Draws a flow chart for an algorithm.
G1.16. Tests an algorithm and debugs errors.
G1.17. Determines the relationship between math and computer science.
G2. Fostering coding skills
G2.1. Explains the basic concepts of programming: The purpose of the program, the purpose of writing the program, and the programming languages.

G2.2. Recognizes the interface and features of the programming tool. Finds and uses open source or free accessible programming tools.
G2.3. Creates the right algorithm to achieve the goals offered in the programming environment. Algorithm processes are performed through simple examples in the programming tool.
G2.4. Explains the structure of linear logic.
G2.5. Develops algorithms that use the linear logic structure.
G2.6. Explains the decision structure and functions.
G2.7. Develops algorithms containing decision structures.
G2.8. Explains the loop structure and functions. The necessity of loop structures for repeated transactions is emphasized.
G2.9. Creates algorithms using loop structure.
G2.10. Extracts the results of the algorithms formed for different structures and extracts their mistakes.

Appendix B

DATUS QUESTIONNAIRE

The below survey items are answered with a 5-level Likert scale. (Strongly Disagree to Strongly Agree) [Sofia da Silva Antunes, 2020]

1. Overall, I am satisfied with how easy it is to use this dashboard.
2. It was simple to use this dashboard.
3. I can effectively complete my work using this dashboard.
4. I am able to complete my goals (tasks) quickly using this dashboard.
5. I am able to efficiently complete my goals (tasks) using this dashboard.
6. I feel comfortable using this dashboard.
7. It was easy to learn to use this dashboard.
8. I believe I became productive quickly using this dashboard.
9. Whenever I make a mistake using the dashboard, I recover easily and quickly.
10. The information (on-screen messages) provided with this dashboard is clear.
11. It was easy to find the information I needed.
12. The information displayed in the dashboard is easy to understand.
13. The information displayed in the dashboard is effective in helping me complete the tasks and scenarios.
14. The organization of the information on the dashboard is clear.
15. The interface of this dashboard is pleasant.
16. I like using the interface of this dashboard.

17. This dashboard has all the functions and capabilities I expect it to have.

18. Overall, I am satisfied with this dashboard.

19. Data on the dashboard is easy to read.

20. Visual encoding of data is consistent throughout the dashboard.

Appendix C

KART-ON UX QUESTIONNAIRE

Merhaba,
Biz bu çalışmada , KARTON programlama uygulaması ile yapacağımız etkinlikler hakkında ne düşündüğünü öğrenmek ve senin cevaplarına göre etkinliğimizi geliştirmek istiyoruz. Çalışmaya katılmaya karar vermek sana kalmış. Eğer katılmak istersen ama sonradan vazgeçersen de her zaman çalışmayı yarıda kesip bırakabilirsin.

- Doğum Yılın:

- Cinsiyetin:

- Okulunda programlama konusunda eğitim aldın mi?
  Evet **Hayır**
- Okulunda Scratch veya benzeri (Mblock, Arduino, Microbit vb.) bir programlama dili kullandın mı?
  Evet **Hayır**
- Daha önceden okul dışında kodlama öğreten bir eğitim programına katıldın mı?
  Evet **Hayır**
- Daha önce, bugün yaptığımız çalışmaya benzer bir atölyeye katıldın mı?
  Evet **Hayır**

Aşağıdaki teknolojik aletleri günde kaç saat kullanıyorsun? *

	Hiç	1-2 saat	3-4 saat	5-6 saat	7-8 saat	8 'den fazla
Bilgisayar	○	○	○	○	○	○
Telefon	○	○	○	○	○	○
Tablet	○	○	○	○	○	○

Aşağıdaki teknolojik aletlerden evinizde var mı? *

	Yok	Kendime ait var	Aile bireylerimin var
Bilgisayar	☐	☐	☐
Akıllı telefon	☐	☐	☐
Tablet	☐	☐	☐

Genel olarak bilgisayarda neler yapıyorsun? (Video izleme, oyun oynama, ders çalışma)

	Kesinlikle Katılıyorum	Katılıyorum	Kararsızım	Katılıyorum	Kesinlikle Katılmıyorum
Teknoloji (örneğin bilgisayar, telefon, tablet) kullanmaktaki başarıma güveniyorum.					
Ben teknolojiyi iyi kullanan bir insanım.					
Teknoloji ile aram iyidir.					

	Kesinlikle Katılıyorum	Katılıyorum	Kararsızım	Katılıyorum	Kesinlikle Katılmıyorum
Uygulamada aklıma gelen programları rahatlıkla yazabildim.					
Uygulamayı kullanma sırasında keyif aldım.					
Uygulamayı kullanmayı hızlıca öğrendim.					
Uygulamayı kullanmak eğlenceliydi.					
Uygulamayı nasıl kullanacağım açıkça belliydi.					
Uygulamada verilen görevler anlaşılırdı.					

	Kesinlikle Katılıyorum	Katılıyorum	Kararsızım	Katılıyorum	Kesinlikle Katılmıyorum
Uygulamada verilen görevler çok zordu.					
Peki,eğer zorlandığın bir yer olduysa en zorladığın görevi bizimle paylaşır mısın?					
Uygulamayı kullanırken kafamın karıştığı yerler oldu.					
Peki, eğer kafanı karıştıran bir yer olduysa en çok kafanı karıştıran yeri bizimle paylaşır mısın?					
Uygulamadaki görevlerin sonundaki çıktılar eğlenceliydi.					
Peki, eğer eğlenceli bulduğun bir yer olduysa en eğlendiğin görevi bizimle paylaşır mısın?					

	Kesinlikle Katılıyorum	Katılıyorum	Kararsızım	Katılıyorum	Kesinlikle Katılmıyorum
Teknoloji ve programlama hakkında daha fazla konu öğrenmek isterim.					
Teknolojik aletlerle çalışmak ilgimi çekiyor.					
Teknolojinin neler yaptığını ve yapabileceğini öğrenmek çok eğlenceli.					
Teknoloji ile ilgili ders alma fikrini seviyorum.					
Benim için teknoloji hakkında bilgi olmak önemlidir.					

	Kesinlikle Katılıyorum	Katılıyorum	Kararsızım	Katılıyorum	Kesinlikle Katılmıyorum
Sınıf arkadaşlarımla beraber çalışmayı ve işbirliği yapmayı seviyorum.					
Sınıf arkadaşlarımla kesinlikle grup çalışması yapmayı isterim.					
Uygulamadaki görevleri grup çalışması olarak yapmak öğrenmeme yardım etti.					
Kullandığımız uygulama ben ve arkadaşlarım arasında işbirliğine izin veriyor.					
Uygulama, programlama aktiviteleri sırasında iletişim kurmaya olanak veriyor.					
Uygulama, gerçekleştirdiğimiz programlama aktiviteleri dışında da iletişim kurmaya olanak veriyor.					

Appendix D

MEB APPROVAL FOR KART-ON UX STUDY

T.C.
İSTANBUL VALİLİĞİ
İl Milli Eğitim Müdürlüğü

Sayı : 59090411-20-E.21380587 09/11/2018
Konu : Anket ve Araştırma İzin Talebi

VALİLİK MAKAMINA

İlgi: a) Koç Üniversitesinin 19.10.2018 tarih ve 51 sayılı yazısı.
b) MEB. Yen. ve Eğ. Tk. Gn. Md. 22.08.2017 tarih ve 12607291/ 2017/25 No'lu Gen.
c) Milli Eğitim Araştırma ve Anket Komisyonunun 08.11.2018 tarihli tutanağı.

Koç Üniversitesi Mühendislik Fakültesi Bilgisayar Mühendisliği Bölümünde görev yapan Prof. Dr. Metin SEZGİN'in **"Erken Dönem Programlama Eğitimi İçin Akıllı Arayüzler"** konulu araştırma çalışması kapsamında, ilimiz genelinde bulunan ilkokullarda öğrenim gören öğrencilere; katılımcı deneyim formu, bilgisayarca düşünme anketi, mekansal yönelme anketi, londra kulesi resimleri testi ve matis düşünme testini uygulama istemi hakkındaki ilgi (a) yazı ve ekleri Müdürlüğümüzce incelenmiştir.

Araştırmacının söz konusu talebi; bilimsel amaç dışında kullanılmaması, **uygulama sırasında bir örneği müdürlüğümüzde muhafaza edilen mühürlü ve imzalı veri toplama araçlarının kurumlarımıza araştırmacı tarafından ulaştırılarak uygulanılması, katılımcıların gönüllülük esasına göre seçilmesi, araştırma sonuç raporunun müdürlüğümüzden izin alınmadan kamuoyuyla paylaşılmaması koşuluyla, okul idarelerinin denetim, gözetim ve sorumluluğunda, eğitim-öğretimi aksatmayacak şekilde ilgi (b) Bakanlık emri esasları dâhilinde uygulanması, sonuçtan Müdürlüğümüze rapor halinde (CD formatında) bilgi verilmesi kaydıyla** Müdürlüğümüzce uygun görülmektedir.

Makamlarınızca da uygun görülmesi halinde olurlarınıza arz ederim.

Levent YAZICI
İl Milli Eğitim Müdürü

Ek:
1- Genelge.
2- Komisyon Tutanağı.

OLUR
09/11/2018

Nihat NALBANT
Vali a.
Vali Yardımcısı

Milli Eğitim Müdürlüğü Binbirdirek M. İmran Öktem Cad.
No:1 Eski Adliye Binası Sultanahmet Fatih/İstanbul
E-Posta: sgb34@meb.gov.tr

A. BALTA VHKİ
Tel: (0 212) 455 04 00-239
Faks: (0 212)455 06 52

Bu evrak güvenli elektronik imza ile imzalanmıştır. https://evraksorgu.meb.gov.tr adresinden 789d-2311-34d3-8482-e4b0 kodu ile teyit edilebilir.

Appendix E

CONSENT FORM FOR KART-ON UX STUDY

Bilgilendirilmiş Gönüllü Olur Formu (BGOF)

Koç Üniversitesi Mühendislik Fakültesi öğretim üyesi Doç Dr. Metin Sezgin tarafından yürütülen, Koç Üniversitesi Etik Kurulları'nın *2018.206.IRB3.144* sayılı onayı ile izin verilen, *Erken Dönem Programlama Eğitimi için Akıllı Arayüzler* başlıklı araştırmaya katılımınız rica olunmaktadır.
Bu araştırmaya tamamen kendi iradenizle, herhangi bir zorlama veya mecburiyet olmadan gönüllü olarak katılımınız esastır. Lütfen aşağıdaki bilgileri okuyunuz ve katılmaya karar vermeden önce anlamadığınız herhangi bir husus varsa çekinmeden sorunuz.

ÇALIŞMANIN AMACI (Neden böyle bir araştırma yapmaya gerek duyuldu?)

Bilgisayar okuryazarlığı ve kodlama becerileri dünya çapında önem kazanmaktadır. Birçok gelişmiş ülkede kodlama eğitimi üniversite öncesinde, ilk ve orta öğretimde başlamaktadır. Dünya genelinde de programlama eğitimini erken yaşlara çeken bir yöneliş mevcuttur. Akademik çalışmalar, programlama eğitiminin temel eğitimin erken evrelerinde verilmesinin öğrencilerin bilişsel açısından gelişiminde önemli olduğunu göstermektedir. Bu doğrultuda, erken yaşta programlama eğitimini destekleyen bir akıllı uygulama olan "Kart-ON" uygulamasını geliştirdik. Kart-ON programlama eğitimine başlangıç için kullanılacak bir programlama dili ve bu dilin çıktılarını derleyebilecek bir mobil uygulamadır.
Gerçekleşecek olan çalışmada amacımız geliştirdiğimiz uygulamanın kullanılabilirliğini ve teknoloji geliştirmeye olan ilgilinin değişimini ölçmektir.

PROSEDÜRLER

Bu çalışmaya çocuğunuzun gönüllü katılmasını istemeniz halinde yürütülecek çalışmalar şöyledir:

Çocuğunuz arkadaşlarıyla birlikte 10 ile 12 kişilik bir grup ile bu çalışmaya katılacaktır. Çalışmada "Kart-ON" uygulamasındaki bir aktivite öğrencilerle birlikte yapılacaktır. Bu aktiviteler çocuğunuzun yaşına uygun olarak, kodlama öğretme amacıyla tasarlanmıştır. Uygulamayı kullanırken bazı veriler kaydedilecektir. Bu veriler uygulama ile geçirilen süre, uygulama ile çekilen resimler, uygulama esnasında kullanılan jestler, ve uygulamanın uzamsal konumudur. Daha sonra uygulamamızı nasıl buldukları, ve nasıl geliştirebileceğimiz hakkında 15-20 dakika sürecek olan bir anketi dolduracaklardır.

OLASI RİSKLER VE RAHATSIZLIKLAR

Çalışmamızın çocuklar üzerinde olası bir riski yoktur.
TOPLUMA VE/VEYA GÖNÜLLÜLERE OLASI FAYDALARI

Bu araştırmanın sonucu "Kart-ON" uygulamasının gelişimine ve hazırlayacağımız müfredatın gelişimine katkı sağlayacaktır. Bu sayede araştırmanın gelecekte tasarlanacak olan eğitim müfredatlarına ve/veya müdahale programlarına yardımcı olması beklenmektedir.

GİZLİLİK
Bu çalışmayla bağlantılı olarak elde edilen ve sizinle özdeşleşmiş her bilgi gizli kalacak, 3. kişilerle paylaşılmayacak ve yalnızca sizin izniniz ile ifşa edilecektir.
Çocuğunuzun bizim vermiş olduğumuz testlere verdiği cevaplar ve uygulamadan alınan veriler şifreli bilgisayarlarda korunacaktır. Öğrencilerin isimleri çalışma bitiminde tüm bilgisayarlardan silinecektir ve araştırmacılardan başka kimseyle paylaşılmayacaktır.

KATILIM VE AYRILMA

Bu çalışmanın içinde olmak isteyip istemediğinize tamamen kendi iradenizle ve etki altında kalmadan karar vermeniz önemlidir.
Katılmaya karar verdikten sonra, herhangi bir anda sahip olduğunuz herhangi bir hakkı kaybetmeden veya herhangi bir yaptırıma maruz kalmadan istediğiniz zaman ayrılabilirsiniz.

ARAŞTIRMACILARIN KİMLİĞİ
Bu araştırma ile ilgili herhangi bir sorunuz veya endişeniz varsa, lütfen iletişime geçiniz:

MERYEM ŞEYDA ÖZCAN
KOÇ ÜNİVERSİTESİ PSİKOLOJİ BÖLÜMÜ

ALPAY SABUNCUOĞLU
KOÇ ÜNİVERSİTESİ BİLGİSAYAR MÜHENDİSLİĞİ

Yukarıda yapılan açıklamaları anladım. Sorularım tatmin olacağım şekilde yanıtlandı. Dilediğim zaman ayrılma hakkım saklı kalmak koşulu ile bu çalışmaya katılmayı onaylıyorum. Bu formun bir kopyası da bana verildi.

____
Katılımcının Adı-Soyadı

____
Ebeveynin Adı-Soyadı

____________________________ ____________________
Ebeveynin İmzası Tarih

____________________________ ____________________
Araştırmacının İmzası Tarih

Appendix F

CONSENT FORM AND SURVEY FOR CLASSROOM ENGAGEMENT DATASET COLLECTION

Gönüllü Bilgilendirme Formu

Koç Üniversitesi Mühendislik Fakültesi/Bilgisayar Mühendisliği Bölümü öğretim üyesi Doç. Dr. T. Metin Sezgin tarafından yürütülen, Koç Üniversitesi Etik Kurulları'nın 2022.177.IRB3.078 sayılı onayı ile izin verilen, *Video Analizi ile Öğrenci Motivasyon ve İstekliliğinin (Angajman) Otonom Tespiti* başlıklı araştırmaya katılımınız rica olunmaktadır.

Bu araştırmaya tamamen kendi iradenizle, herhangi bir zorlama veya mecburiyet olmadan gönüllü olarak katılımınız esastır. Lütfen aşağıdaki bilgileri okuyunuz ve katılmaya karar vermeden önce anlamadığınız herhangi bir husus varsa çekinmeden sorunuz.

ÇALIŞMANIN AMACI (Neden böyle bir araştırma yapmaya gerek duyuldu?)

Öğrenci motivasyonu ve sınıf içi katılımını (angajman) ölçmek, öğrencinin gelişimini ve gelecek başarısını tahmin etmek için en önemli unsurlardandır. Bu ölçütlerin değerlendirilmesi ancak öğrencinin birçok açıdan incelenmesi ile doğru bir şekilde gerçekleşebilir. Bu çalışmada, eğitim ve yapay zekâ literatürüne yeni ve zengin bir veri setinin kazandırılması hedeflenmiştir. Bu veri setinde, grup halinde fiziksel ve dijital yaratıcı kodlama çalışmalarının video kesitleri sunulacaktır.

PROSEDÜRLER

Bu çalışmaya gönüllü katılmak istemeniz halinde yürütülecek çalışmalar şöyledir:

- Toplamda altmış dakika sürecek bir yaratıcı kodlama eğitimine katılacaksınız. Bu çalışmanın yaklaşık otuz dakikası tablet kullanarak programlama yapmayı, diğer otuz dakikası ise karton, renkli kalemler ve oyun hamuru gibi el işi malzemelerini kullanacağınız aktiviteleri içermektedir.

-
Aktivitenin programlama aşamasında Javascript dilini, özellikle p5.js çizim kütüphanesini kullanacaksınız. Önceden bu dile ve kütüphane ile herhangi bir deneyim olması aranmamaktadır.

-
Eğitim sırasında, yüzünüze doğrudan bakan bir kamera ve grubu farklı açılardan gören kameralar ile video çekilecektir. Bu videolar, sadece araştırmacılar ve bu videolarını kendi araştırmaları için kullanacak, araştırmacıların onayladığı kişiler tarafından görülebilecektir.

OLASI RİSKLER VE RAHATSIZLIKLAR

Çalışmanın olası bir riski ve yaratacağı herhangi bir olumsuz durum öngörülmemiştir.

TOPLUMA VE/VEYA GÖNÜLLÜLERE OLASI FAYDALARI

Toplanacak veri seti ile, öğretmenlerin iş yükünü azaltıcı teknolojilerin geliştirilmesi için önemli bir adım atılacaktır. Güncel veri setleri ile, ancak çevrimiçi platformlar için öğrenci katılımını ölçebilecek teknolojiler geliştirilebilmektedir. Toplanacak veri seti ile eğitim literatürüne uygun, öğrenciyi farklı açılardan, daha doğru değerlendirilebilecek video-temelli sistemler geliştirilebilecektir.

GİZLİLİK

Bu çalışmayla bağlantılı olarak elde edilen ve sizinle özdeşleşmiş her bilgi gizli kalacak, 3. kişilerle paylaşılmayacak ve yalnızca sizin izniniz ile ifşa edilecektir.

Bu çalışmadan elde edilen görüntüler onay vermeniz durumunda, sadece,

(1)
Çalışmayı gerçekleştiren araştırmacılarca kullanılacak,

(2)
Çalışmayı gerçekleştiren araştırmacıların onay verdiği
akademik amaçlı çalışmalardaki diğer araştırmacılarca kullanılabilecektir.

Bulunduğunuz video
kayıtlarının paylaşılmamasını istediğiniz andan itibaren, yüzünüzün yer aldığı
tüm içerik veri setinden çıkarılacaktır.

KATILIM VE AYRILMA

Bu çalışmanın
içinde olmak isteyip istemediğinize tamamen kendi iradenizle ve etki altında
kalmadan karar vermeniz önemlidir.

Katılmaya karar
verdikten sonra, herhangi bir anda sahip olduğunuz herhangi bir hakkı
kaybetmeden veya herhangi bir yaptırıma maruz kalmadan istediğiniz zaman
ayrılabilirsiniz.

* Indicates required question

1. İsim ve soyisim *

____

2. E-posta adresiniz: *

____

3. Yukarıda yapılan açıklamaları anladım. Sorularım tatmin olacağım şekilde yanıtlandı. Dilediğim zaman ayrılma hakkım saklı kalmak koşulu ile bu çalışmaya katılmayı onaylıyorum. *

Mark only one oval.

- Evet
- Hayır

Veri Seti Yayınlama İzni

Bu form, 2022.177.IRB3.078 no'lu etik kurul izni ile gerçekleştirilen, "*Video Analizi ile Öğrenci Motivasyon ve İstekliliğinin (Angajman) Otonom Tespiti*" projesi kapsamında video veri toplama toplama çalışmasına katılmanız üzerine yollanmaktadır. Topladığımız veri araştırmacının gerçekleştirdiği yapay zeka modeli geliştirme çalışmalarında kullanılacaktır. Siz izin vermediğiniz sürece hiçbir veri, araştırmacılar haricinde kimseyle paylaşılmayacaktır.

4. Yukarıdaki yazan açıklamayı okudum ve anladım. *

Mark only one oval.

- Evet
- Hayır

5. Videoların görüntü ve ses içeriklerinin paylaşılmasına izin veriyorum. *

Mark only one oval.

- Evet
- Hayır

6. Videoların fotoğraf serisine dönüştürülmüş anonim formatta paylaşılmasına izin veriyorum. *

Mark only one oval.

- Evet
- Hayır

7. Videoların metin içeriğinin anonim bir şekilde paylaşılmasına izin veriyorum. *

Mark only one oval.

- Evet
- Hayır

8. Yapay zeka modellerinden çıkan yüz, vücut, ses ve metin özelliklerini betimleyen anonim verilerin paylaşılmasına izin veriyorum. *

Mark only one oval.

- Evet
- Hayır

Angajman Anketi

9. Eğitim Durumu *

Mark only one oval.

- Lisans 1-2. sınıf
- Lisan 3-4. sınıf
- Yüksek Lisans
- Doktora

10. Daha önceden yaratıcı kodlama ilgili aktivitelere katıldınız mı? Bu tarz kodlama becerinizi hangi düzeyde tanımlarsınız? *

11. Genel olarak kendinizi dersleri/aktiviteleri takip eden biri olarak mı tanımlarsınız? *

12. Önceki deneyimlerinize göre, öğrenme aşamasında hangi tarz aktiviteler yapmak hoşunuza gider? *

13. Önceki deneyimlerinize göre, öğrenme aşamasında hangi tarz aktiviteler motivasyonunuzu düşürür? *

This content is neither created nor endorsed by Google.

Google Forms

Appendix G

ABSTRACTS OF PUBLISHED AND IN-SUBMISSION ARTICLES

G.1 Kart-ON: Affordable Early Programming Education with Shared Smartphones and Easy-to-Find Materials

Programming education has become an integral part of the primary school curriculum. However, most programming practices rely heavily on computers and electronics which causes inequalities across contexts with different socioeconomic levels. This demo introduces a new and convenient way of using tangibles for coding in classrooms. Our programming environment, Kart-ON, is designed as an affordable means to increase collaboration among students and decrease dependency on screen-based interfaces. Kart-ON is a tangible programming language that uses everyday objects such as paper, pen, fabrics as programming objects and employs a mobile phone as the compiler. Our preliminary studies with children ($n = 16, m_{age} = 12$) show that Kart-ON boosts active and collaborative student participation in the tangible programming task, which is especially valuable in crowded classrooms with limited access to computational devices.

`https://dl.acm.org/doi/abs/10.1145/3379336.3381472`

G.2 Devoloping Affordable Tangible Programming Education Applications Using Mobile Vision

Programming education has become an essential part of the primary and secondary school curriculum. Two main programming environment modalities, web-based visual programming applications and electronic cards, have become popular in curricular activities. Limiting the programming activities around these programming environments restricts education accessibility in socio-economically disadvantaged regions due to the need for an individual computer per student, lack of decent infrastructure, and high electronics prices.

Effective, shared use of smartphones and tablets in programming education can provide equal opportunities. In this scenario, students can code simple drawings and animations using their own materials as tangible programming blocks by employing a single shared phone in the classroom as an interpreter. This article explains our development process of a new tangible programming environment which increases the accessibility of education. We discuss effective inclassroom use of image/text processing practices and transfer learning methods on smartphones.

`https://ieeexplore.ieee.org/abstract/document/9477839/`

G.3 Teaching K-12 Classrooms Data Programming: A Three-Week Workshop with Online and Unplugged Activities

This paper shares our experience in a three-session online workshop using a new web-based data programming environment, Marti. The programming environment uses a card-based programming strategy in both unplugged and online activities. Educators can use the physical cards in a board-game style or use the programming environment's mobile application to scan these cards and render the final visualization on their phones/tablets. The web environment also uses visual draggable cards for programming that can manipulate and visualize data. We used Marti and its offered unplugged activities in three sessions with 12 middle school and 12 high school students, focusing on the data fundamentals, analysis, and visualization. We assert that integrating unplugged-style pseudo-code creation and supporting a similar experience using the available devices have considerable potential for delivering equal and affordable data programming education for all.

`https://arxiv.org/abs/2110.05303`

G.4 Prototyping Products using Web-based AI Tools: Designing a Tangible Programming Environment with Children

A wide variety of children's products such as mobile apps, toys, and assistant systems now have integrated smart features. Designing such AI-powered products with children, the users, is essential. Using high-fidelity prototypes can be a means to reveal children's needs and behaviors with AI-powered systems. Yet, a prototype that can show unpredictable features similar to the final AI-powered product can be expensive. A more manageable and inexpensive solution is using web-based AI prototyping tools such as Teachable Machine. In this work, we developed a Teachable Machine-powered game-development environment to inform our tangible programming environment's design decisions. Using this kind of an AI-powered high-fidelity prototype in the research process allowed us to observe children in a very similar setting to our final AI-powered product and extract design considerations. This paper reports our experience of prototyping AI-powered solutions with children and shares our design considerations for children's self-made tangible representations.

`https://dl.acm.org/doi/abs/10.1145/3535227.3535239`

G.5 Kart-ON: An Extensible Paper Programming Strategy for Affordable Early Programming Education

Programming has become a core subject in primary and middle school curricula. Yet, conventional solutions for in-class programming activities require each student to have expensive equipment, which creates an opportunity gap for low-income students. Paper programming can provide an affordable, engaging, and collaborative in-class programming experience by allowing groups of students to use inexpensive materials and share smartphones. However, current paper-programming examples are limited in terms of language expressivity and generalizability. Addressing these limitations, we developed a paper-programming flow and its variants in different abstraction levels and input/output styles. The programming environments consist of pre-defined tangible programming cards and a mobile application that runs computer vision models to recognize them. This paper describes our educational and technical development process, presents a qualitative analysis of the early user study results and shares our design considerations to help develop wide-reaching paper programming environments.

`https://dl.acm.org/doi/abs/10.1145/3534524`

G.6 Exploring Children's Use of Self-Made Tangibles in Programming

Defining abstract algorithmic structures like functions and variables using self-made tangibles can enhance the usability and affordability of the tangible programming experience by maintaining the input modality and physical interaction throughout the activity and reducing the dependence on electronic devices. However, existing tangible programming environments use digital interfaces to save abstract definitions such as functions and variables, as designing new tangibles is challenging for children due to their limited experience using abstract definitions. We conducted a series of design workshops with children to understand their self-made tangible creation abilities and develop design considerations for tangible computing such as paper programming. This paper reports: (1) Our insights on how students conceptualize and design tangible programming blocks, (2) Design considerations of self-made tangibles to yield higher understandability and memorability, (3) Tests of our design considerations in creating self-made tangibles in real-life coding activities.

`https://arxiv.org/abs/2210.06258`

G.7 Developing a Multimodal Classroom Engagement Analysis Dashboard for Higher-Education Students

Developing learning analytics dashboards (LADs) is a growing research interest as online learning tools have become more accessible in K-12 and higher education settings. This paper reports our *multimodal classroom engagement data analysis* and *dashboard design process* and the resulting engagement dashboard. Our work stems from the importance of monitoring classroom engagement, which refers to students' active physical and cognitive involvement in learning that influences their motivation and success in a given course. To monitor this vital facade of learning, we developed an engagement dashboard using an iterative and user-centered process. We first created a multimodal machine learning model that utilizes face and pose features obtained from recent deep learning models. Then, we created a dashboard where users can view their engagement over time and discover their learning/teaching patterns. Finally, we conducted user studies with undergraduate and graduate-level participants to obtain feedback on our dashboard design. Our paper makes three contributions by (1) presenting a student-centric, open-source *dashboard*, (2) demonstrating *a baseline architecture* for engagement analysis using *our open-access data*, and (3) presenting *user insights and design takeaways* to inspire future LADs. We expect our research to guide the development of tools for novice teacher education, student self-evaluation, and engagement evaluation in crowded classrooms.

`https://doi.org/10.1145/3593240`

G.8 Towards Building Child-Centered Machine Learning Pipelines: Use Cases from K-12 and Higher-Education

Researchers and policy-makers have started creating frameworks and guidelines for building machine-learning* (ML) pipelines with a human-centered lens [Amershi et al., 2019, Laura Musikanski et al., 2018]. Machine Learning pipelines stand for all the necessary steps to develop ML systems (e.g., developing a predictive keyboard) [Paleyes et al., 2023]. On the other hand, a child-centered focus in developing ML systems has been recently gaining interest as children are becoming users of these products [UNESCO, 2021]. These efforts dominantly focus on children's interaction with ML-based systems. However, from our experience, ML pipelines are yet to be adapted using a child-centered lens. In this paper, we list the questions we ask ourselves in adapting human-centered ML pipelines to child-centered ones. We also summarize two case studies of building end-to-end ML pipelines for children's products.

`https://arxiv.org/abs/2304.09532`

G.9 Multimodal Group Activity Dataset for Classroom Engagement Level Prediction

Classroom engagement is a key factor in predicting students' future success and motivation. Thus, analyzing student engagement and guiding students and teachers has been an ongoing research interest in learning analytics. Recent research on deep learning-based engagement analysis demonstrated a promising development. However, existing research lacks presenting the multimodal nature of engagement in the real-life classroom setting. To accelerate the research in this field, we collected a new dataset that includes approximately eight hours of audiovisual recordings of a group of students and their self-evaluation scores for classroom engagement. We developed baseline face-based and group-activity-based image and video recognition models. Our image models yield 45-85% test accuracy with face-area inputs on person-based classification task. Our video models achieved up to 71% test accuracy on group-level prediction using group activity video inputs. In this paper, we explore our end-to-end human-centered engagement analysis pipeline from data collection to model development. We contribute to human-AI interaction research by providing a new audiovisual dataset for classroom engagement and baseline machine learning architectures and open-source models.

https://arxiv.org/abs/2304.08901

G.10 Developing an Open-Source Middle School Programming Environment and its Curriculum for an Affordable Education

Programming education has become a core component of the K-12 curriculum in many countries. We developed our programming application, Kart-ON, to increase access to affordable programming education by supporting algorithm building with tangibles, and promoting mobile device sharing between groups in the classroom. In a typical scenario, while using our curriculum, students first create algorithms working in groups using paper programming cards and custom tangible objects, then use a shared smartphone or tablet to scan the cards and run the output. The programming environment was developed in 3 stages: In Phase I, we developed three different versions of our programming environment in terms of programming commands and input/output modalities. In Phase II, a user experience (UX) study was conducted with ninety-one students to compare three groups regarding measuring the usability, collaboration, and enjoyability of different versions. Results showed that using simplified input blocks and getting real-time digital outputs have a positive impact on the student experience. In Phase III, we conducted an ideation workshop with teachers and other education stakeholders and developed our curriculum based on the findings of this workshop and UX results. This paper makes three contributions to educational technology research by providing (i) the in-class working flow of our affordable, tangible, open-sourced programming environment, (ii) the user experience study analysis that interpreted to design takeaways and (iii) our suggested open-source new middle school programming education curriculum.

In Submission