Interactive Machine Teaching with and for Novices

Téo Sanchez

université
PARIS-SACLAY

Interactive Machine Teaching with and for Novices

Enseignement machine interactif avec et pour les novices

Thèse de doctorat de l'université Paris-Saclay

École doctorale n°580 : Sciences et technologies de l'information et de la communication (STIC)
Spécialité de doctorat : Informatique
Graduate School : Informatique et sciences du numérique
Référent : Faculté des sciences d'Orsay

Thèse préparée dans l'unité de recherche **Laboratoire Interdisciplinaire des Sciences du Numérique** (Université Paris-Saclay, CNRS, Inria), sous la direction de **Wendy E. MACKAY**, directrice de recherche, et le co-encadrement de **Baptiste CARAMIAUX**, chargé de recherche au CNRS.

Thèse soutenue à Paris-Saclay, le 20 juin 2022, par

Téo SANCHEZ

Composition du jury

Michèle SEBAG Directrice de recherche, Université Paris-Saclay	Présidente
Antti OULASVIRTA Professeur, Aalto University	Rapporteur & Examinateur
Gonzalo RAMOS Chargé de recherche, Microsoft	Rapporteur & Examinateur
Albrecht SCHMIDT Professeur, Ludwig Maximilian University of Munich	Rapporteur & Examinateur
Simone STUMPF Maîtresse de conférences, University of Glasgow	Examinatrice
Baptiste CARAMIAUX Chargé de recherche, Sorbonne Université	Co-encadrant & Examinateur
Wendy E. MACKAY Directrice de recherche, Université Paris-Saclay	Directrice de thèse

THESE DE DOCTORAT
NNT : 2022UPASG055

université PARIS-SACLAY

ÉCOLE DOCTORALE
Sciences et technologies de l'information et de la communication (STIC)

Titre : Enseignement machine interactif avec et pour les novices
Mots clés : Interaction Humain-Machine, Apprentissage machine, Évaluations centrées sur l'humain

Résumé : Les algorithmes d'apprentissage machine déployés dans la société ou la technologie offrent généralement aux utilisateurs aucune prise sur la manière dont les modèles d'apprentissage sont optimisées à partir des données. Seuls les experts conçoivent, analysent et optimisent les algorithmes d'apprentissage automatique.
À l'intersection de l'Interaction Humain-Machine (IHM) et de l'apprentissage machine, le domaine de l'apprentissage automatique interactif vise à intégrer l'apprentissage automatique dans des pratiques existantes. L'enseignement machine interactif (Interactive Machine Teaching), en particulier, cherche à impliquer des utilisateurs non experts en tant qu'enseignant de la machine afin de les autonomiser dans le processus de construction de modèles d'apprentissage. Ces utilisateurs pourraient profiter de la construction de modèles d'apprentissage pour traiter et automatiser des tâches sur leurs propres données, conduisant à des modèles plus robustes et moins biaisés pour des problèmes spécialisés.
Cette thèse adopte une approche empirique sur l'enseignement machine interactif en se concentrant sur la façon dont les utilisateurs développent des stratégies et comprennent les systèmes d'apprentissage machine interactifs à travers l'acte d'enseigner. Cette recherche fournit deux études utilisateurs impliquant des participants en tant qu'enseignant de classificateurs d'images utilisant des réseaux de neurones artificiels appris par transfert. Ces études se concentrent sur ce que les utilisateurs comprennent du comportement du modèle ML et sur la stratégie qu'ils peuvent utiliser pour le "faire fonctionner". La seconde étude se concentre sur la compréhension et l'utilisation de deux types d'incertitude : l'incertitude aléatorique, qui traduit l'ambiguïté, et l'incertitude épistémique, qui traduit la nouveauté. Je discute de l'utilisation de l'incertitude et de l'apprentissage actif (Active Learning) comme outils pour l'enseignement machine interactif. Enfin, je présente mes collaborations artistiques et adopte une approche réflexive sur les obstacles et les opportunité de développement de l'apprentissage automatique interactif pour l'art.

Je soutiens que les utilisateurs novices développent différentes stratégies d'enseignement qui peuvent évoluer en fonction des informations obtenues tout au long de l'interaction. Les stratégies d'enseignement structurent la composition des données d'entraînement et affectent la capacité des utilisateurs à comprendre et à prédire le comportement de l'algorithme.

En plus de permettre aux gens de construire des modèles d'apprentissage automatique, l'enseignement machine interactif présente un intérêt pédagogique en favorisant les comportements d'investigation, renforçant les connaissances des novices en apprentissage machine.

université PARIS-SACLAY
ÉCOLE DOCTORALE
Sciences et technologies de l'information et de la communication (STIC)

Title : Interactive Machine Teaching with and for novices
Keywords : Human-Computer Interaction, Machine Learning, Human-centered evaluations

Abstract :
Machine Learning algorithms in society or interactive technology generally provide users with little or no agency with respect to how models are optimized from data. Only experts design, analyze, and optimize ML algorithms.
At the intersection of HCI and ML, the field of Interactive Machine Learning (IML) aims at incorporating ML workflows within existing users' practices. Interactive Machine Teaching (IMT), in particular, focuses on involving non-expert users as "machine teachers" and empowering them in the process of building ML models. Non-experts could take advantage of building ML models to process and automate tasks on their data, leading to more robust and less biased models for specialized problems.
This thesis takes an empirical approach to IMT by focusing on how people develop strategies and understand interactive ML systems through the act of teaching. This research provides two user studies involving participants as teachers of image-based classifiers using transfer-learned artificial neural networks. These studies focus on what users understand from the ML model's behavior and what strategy they may use to "make it work." The second study focuses on machine teachers' understanding and use of two types of uncertainty : aleatoric uncertainty, which conveys ambiguity, and epistemic uncertainty, which conveys novelty. I discuss the use of uncertainty and active learning in IMT. Finally, I report artistic collaborations and adopt an auto-ethnographic approach to challenges and opportunities for developing IMT with artists.
I argue that people develop different teaching strategies that can evolve with insights obtained throughout the interaction. People's teaching strategies structure the composition of the data they curated and affect their ability to understand and predict the algorithm behavior.
Besides empowering people to build ML models, IMT can foster investigative behaviors, leveraging peoples' literacy in ML and artificial intelligence.

Acknowledgments

To my supervisor **Baptiste Caramiaux**, I am deeply grateful for the constant and unwavering support he gave me before, during, and after this thesis. Baptiste taught me most of what I know in HCI, ML, and research ethics, as well as made me discover novel forms of art. He made me feel like his equal from the very beginning. Exploring and learning together on this topic was a delight. As I wish to pursue academia, his research approach, humility and open-mindedness will always be examples to follow.

Likewise, I am sincerely grateful to my thesis director **Wendy Mackay**, who taught me the fundamentals of situated interaction, design techniques, experimental design, writing skills, and ethics, among many others. She shares her passion for research with inexhaustible energy. Her workshops, boot camps, and hackathons are some of the best moments of my doctoral training. She also knew how to find the right words to push me through stressful deadlines.

I had the chance to collaborate with amazing researchers without whom this research would not exist: **Jules Françoise**, whose research on Marcelle simplified the life of many researchers (especially mine); **Frédéric Bevilacqua** who welcomed me in the ELEMENT project, opening new perspectives to my work; and **Pierre Thiel**, a talented master student whose collaboration has been extremely fruitful and fun.

I could not have wished for a better thesis jury than **Antti Oulasvirtta**, **Gonzalo Ramos**, **Albrecht Schmidt**, **Simone Stumpf**, and **Michèle Sebag**. After so many virtual conferences during the covid crisis, these two hours of deep and thoughtful discussion were an absolute gift! Thank you.

I do not forget members of the **Traces** association and the **Projet Siscode**: **Matteo Merzagora**, **Aude Ghilbert**, **Paul Boniface**, and **Arnaud Malher**. Thanks to **Gianni Franchi** for his thoughtful advice on deep learning uncertainty, and **Calvin Peck** for proofreading my thesis and last-minute encouragement.

Teaching was a central experience in my doctoral work. Despite the considerable teaching load of full-time ATER, I **always** enjoyed being with students and trying to get the best out of them. I also met some incredible professors: **Nicolas Thiéry**, **Fanny Pouyet**, **Adeline Pierrot**, **Kim Nguyen**, **Viviane Pons**, and

Florent Hivert among others. Students in Paris-Saclay are very lucky to have a team of such dedicated professors. My gratitude especially goes to **Nicolas Thiéry**, who was kind of a third mentor to me and an inspiring model to follow. I will not forget his enthusiastic, benevolent, and demanding attitude toward students and colleagues.

I also thank all my colleagues, met during my long stay in *LISN* (almost 4 years) and *ISIR*: Thank you **Camille Gobert**, for your kindness and support back in ENS to the thesis defense; **Han Han** for bringing a lot of fun at work and at home; **Liz Walton** for sharing our doubts and supporting each other; my H05 mates **Aleks Vereschak**, **Clara Rigaud**, **Vaynee Sungeelee**, and **Antoine Loriette**; thank you **Miguel Renom** for hosting me and for your humor; **Viktor Gustafsson** to make me discover evolving narratives in video games; **Yi Zhang** for the memorable catacomb excursion. I am grateful to the impressive previous Ph.D. students I met, **Jean-Philippe Rivière**, **Abby Liu**, **German Leiva**, **Carla Griggio**, **Stacy Hsueh**, **Philip Tchernavskij**, **Jessalyn Alvina**, **Michael Wessely**, and **Nolwenn Maudet**. All the best and gratitude to the students I am leaving: Thank you **Martin Tricaud** for your personal touch during the graduation ceremony; **Tove Grimstad Bang**, **Capucine Nghiem**, and **Léa Paymal** for this memorable evening in Teddy's bar; my office partner **Wissal Sahel**; van-life and mixed reality expert **Arthur Fages**; my hackathon teammate **Junhang Yu**; thank you **Alexandre Battut**, **Anna Offenwanger**, **Romane Dubus**, and **Flavien Lebrun**. Thank you to all researchers I met there whose work and comments contributed to my doctoral training: **Michel Beaudouin-Lafon**, **Gilles Bailly**, **Sarah Fdili-Alaoui**, **Fanis Tsandilas**, **Nicolas Taffin**, **Janin Koch**, and **Julien Gori**.

Many thanks to the **CESFO restauration team**, for their sympathy, professionalism, and cooking skills. They have an essential role in the well-being of the workers on the campus.

I am extremely grateful for the unconditional support of my family: my parents, **Patrick** and **Sylvie**, my brother **Pablo**, my grandfather **Jean-Pierre**, and **Pierrot**. Thank you all my friends, especially **Armine** and **Robert** for the warmth of your welcome and the numerous dinner we shared at your place.

Géraldine, for being by my side in moments of panic and joy. I hope to bring you the same support to help you become a great surgeon. Merci du fond du coeur.

Contents

List of Figures

List of Tables

Résumé étendu en français

Les systèmes d'apprentissage automatique, en particulier les réseaux neuronaux profonds, peuvent s'attaquer à un nombre toujours plus grand de tâches complexes, en utilisant des données annotés. Ces algorithmes d'apprentissage automatique sont désormais omniprésentes dans les domaines générant des données, tels que la science, la finance, l'ingénierie, la médecine, le droit, l'administration et l'art.

Cependant ces algorithmes n'offrent peu ou pas de prise sur la manière dont ils sont optimisées. La conception, l'analyse et l'optimisation des algorithmes ML sont principalement le fait d'experts, tels que des ingénieurs ou des chercheurs, qui utilisent des outils de programmation, d'analyse et de visualisation hors de portée du grand public.

À l'intersection de l'Interaction Humain-Machine (IHM) et de l'apprentissage machine, l'enseignement machine interactif (Interactive Machine Teaching) cherche à impliquer des utilisateurs non experts en tant qu'enseignants afin de les autonomiser dans le processus de construction de modèles d'apprentissage. Ces utilisateurs pourraient profiter de la construction de modèles d'apprentissage pour traiter et automatiser des tâches sur leurs données, ce qui pourrait conduire à des modèles plus robustes et moins biaisés pour des problèmes spécialisés.

Cette thèse adopte une approche empirique sur l'enseignement machine interactif en se concentrant sur la façon dont les utilisateurs développent des stratégies et comprennent les systèmes d'apprentissage machine interactifs à travers l'acte d'enseigner.

Le premier chapitre introduit ce travail et souligne l'importance de la capacité des novices à entraîner leurs propres algorithmes d'apprentissage.

Le chapitre 2 définit d'abord l'apprentissage automatique interactif et les recherches existantes. Il présente ensuite l'enseignement machine interactif (IMT) et ses spécificités. Enfin, le chapitre présente Marcelle, une boîte à outils pour la composition de flux de travail et d'interfaces en apprentissage automatique interactif. Cet outil a servi à menner les recherches présentées dans cette thèse et je présente en quoi il et

pourrait aider à la conception de systèmes enseignables.

Le chapitre 3 se concentre sur la manière dont les novices utilisent les algorithmes d'apprentissage, sur ce qu'ils comprennent de leur comportement et la stratégie qu'ils utilisent pour les faire apprendre. La première section présente le contexte de vulgarisation scientifique dans lequel cette recherche est ancrée. Les sections suivantes présentent ensuite une quasi-expérience (observation structurée) dans laquelle les participants ont effectué des tâches d'enseignement de la machine à l'aide d'un protocole de réflexion à voix haute.

Le chapitre 4 adopte une approche centrée sur l'humain pour l'évaluation de l'incertitude dans l'apprentissage profond. Il explore comment les deux types d'incertitude—aléatorique et épistémique—peuvent aider les utilisateurs novices à comprendre les forces et les faiblesses d'un classificateur d'image dans un scénario d'apprentissage interactif par machine (IML).

Le chapitre 5 présente deux collaborations artistiques qui ont donné lieu à deux installations impliquant des algorithmes d'apprentissage automatique. Il une perspective réflexive et discute des défis confrontés pour utiliser l'apprentissage automatique dans des projets artistiques contrastés.

Le chapitre 6 discute des stratégies d'enseignement des participants ainsi que le rôle de l'incertitude dans une tâche d'enseignement. Enfin, j'aborde les implications socioculturelles de cette recherche, comme l'utilisation de l'enseignement machine interactif pour la pédagogie.

Je soutiens que les utilisateurs novices développent différentes stratégies d'enseignement qui peuvent évoluer en fonction des informations obtenues tout au long de l'interaction. Les stratégies d'enseignement structurent la composition des données d'entraînement et affectent la capacité des utilisateurs à comprendre et à prédire le comportement de l'algorithme.

En plus de permettre aux gens de construire des modèles d'apprentissage automatique, l'enseignement machine interactif présente l'avantage pédagogiques peut favoriser les comportements d'investigation, renforçant les connaissances implicites des novices en apprentissage machine.

Chapter 1

Introduction

This thesis investigates peoples' interactions and understanding of interactive machine learning systems when placed in the role of machine teachers. Interactive machine teaching (IMT) systems are specifically designed to leverage barriers to creating machine learning (ML) models by exploiting humans' natural ability to teach. [Ramos et al., 2020]. This work mainly focuses on peoples' understanding and appropriation of image-based IMT systems involving users in creating their own data. Observing end-users teaching IMT systems can simultaneously shed light on how they reason, interact and learn about teachable interactive systems, which is essential with systems that convey complex notions such as predictive uncertainty. This thesis analyzes evidence that users' teaching strategies affect their understanding of ML models.

1.1 Context

Since machines have demonstrated behaviors that simulate cognitive abilities, attention has been placed on improving these abilities. The first "intelligent" machines were proofs of concept, but their inventors already anticipated that it would not take long for them to surpass human intelligence, leaving us all as spectators of a new kind of supra-intelligence [Grudin, 2009]. History proved them wrong, revealing a way less linear (or exponential) evolution than expected. Instead, artificial intelligence has developed through waves and winters, in which different paradigms (symbolic or connectionist) succeeded one another [Cardon et al., 2018]. More importantly, the goal of making machines more intelligent now coexist with the goal of augmenting human intelligence through interaction and partnership.

Nowadays, AI mainly revolves around Machine Learning (ML) algorithms that automatically improve on a task based on experience accu-

mulated from data [Jordan and Mitchell, 2015]. They produce models that can be used to automate many of our activities by first modeling phenomena or concepts from real-world data and then predicting future outcomes with new data. ML algorithms increasingly take on governance roles in public and private domains, implying considerable ethical implications. ML algorithms are widely used for facial recognition in streets and airports in certain countries [wik, 2004b], they assist decision-making in the medical field [Bhatt et al., 2021a, Bakator and Radosav, 2018, Shen et al., 2017] and justice field [wik, 2004a], and are developed in the automotive industry for autonomous driving [Muhammad et al., Grigorescu et al.]. Deep Learning, in particular, involves artificial neural networks and demonstrates the best performances in complex tasks using raw data, such as recognizing speech, identifying elements in images, and generating realistic sound and images. The high learning capacity of these models also requires a large amount of data in the form of vectors, which are the intermediate representations on which ML algorithms learn from the world. This increasing need for data shaped different humans roles toward ML.

On the ML expert side, researchers, engineers, developers, and data scientists are in charge of designing new learning algorithms, analyzing data, training and evaluating ML models on these data, and deploying them in interactive technology used in society. On the other side of the expertise scale, the general public mainly provide labeled data, with or without their knowledge or consent[1]. For instance, *microworkers* or *turkers*[2] are remotely located workers that perform small units of work on tasks for which no efficient algorithm has been designed yet. By labeling and preparing data, micro-workers are also at the source of the virtual assembly line of ML models[3]. Hence, ML systems are trained on peoples' data, but most of them, including subject-matter experts, which are experts in other disciplines than computer science or ML, are excluded from controlling how ML models are trained or deployed.

Non-experts in ML could take advantage of building ML models to process and automate tasks on customized data. Involving people other than ML experts in the creation of ML models could potentially result in more robust and less biased models for specialized problems as well as the development of new communities of practice. For example, a specialized medical doctor could train an ML model to classify their own scan images without the help of an ML specialist but with appropriate interactive tools. The model built could be shared with students to support their diagnoses in their absence.

[1] A well-known example is CAPTCHAs (Completely Automated Public Turing test to tell Computers and Humans Apart), which are test questions used on many platforms to determine whether users "are not a robot." Some CAPTCHA tasks are a pretext for collecting tagged images for the growing industry of autonomous driving for example.

[2] *Turker* refers to the Amazon Mechanical Turk platform, a crowdsourcing website for businesses to hire remotely located micro-workers. The platform's name refers to an 18th-century chess-playing fake automaton made by Wolfgang von Kempelen that toured Europe.

[3] A detailed cartography of the human labor division and interaction involved in AI can be found on https://anatomyof.ai/.

This dissertation is interested in understanding the role of **machine teacher** [Ramos et al., 2020], i.e. people involved in the process of building ML models, with a specific focus on non-experts in ML from the general public or among subject-matter experts.

Interactive Machine Teaching

Interactive Machine Teaching (IMT) is a research field that aims at designing interactive ML systems designed to leverage peoples' inherent teaching abilities [Ramos et al., 2020]. IMT systems are intended to enable users without scientific or technological expertise to create ML models with interactive ML systems.

IMT is defined in contrast to the traditional ML field. While ML focuses on improving models given data, IMT aims at *«making human teachers more productive at building ML models, given a learning algorithm»* [Simard et al., 2017]. IMT focuses on *«extracting knowledge from people rather than extracting knowledge from data»* [Ramos et al., 2020]. Hence, the corresponding research seeks to improve the dialog fluency between humans and interactive ML systems.

Interactive Machine Teaching should not be mistaken for Machine Teaching in the realm of the computational learning theory [Shinohara and Miyano, 1991, Zhu, 2015, Zhu et al., 2018]. In this context, Machine Teaching (MT) is framed as the inverse problem of ML in the sense that in ML, the source is the training data, and the target is the trained model. In MT, the source is the trained model, and the target is an optimal set of examples leading to this trained model. There is no dialog between a human teacher and a learning machine, although MT finds some uses in intelligent tutoring systems [Koedinger et al., 2013]. The tutoring system tries to model the student knowledge in order to find the optimal examples [Patil et al., 2014].

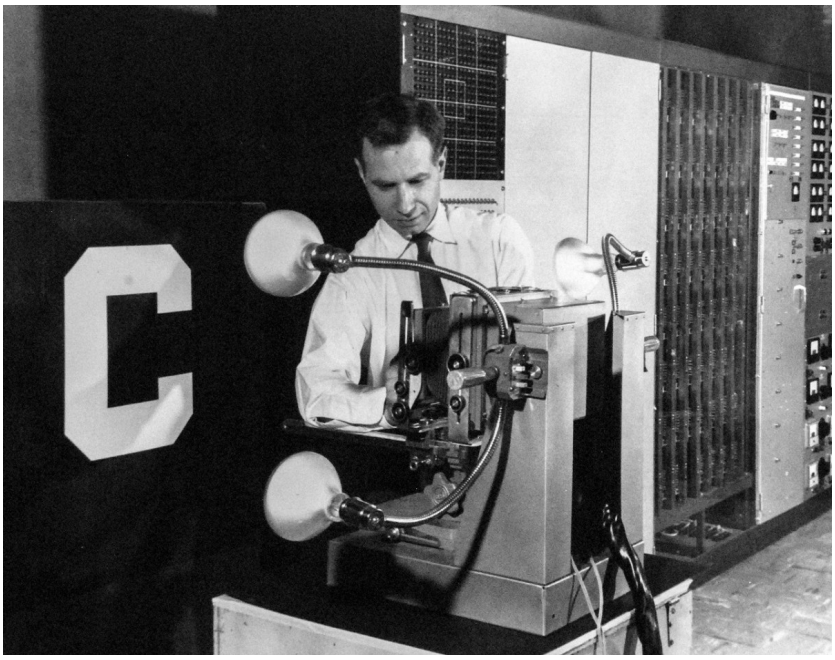


Figure 1.1. Frank Rosenblatt and the Perceptron Mark 1 in the 1960s. Source: Cornell university website, wikimedia creative common.

The idea of a human teaching a machine has appeared under different names and forms in the HCI literature [Ware et al., 2001]. Maybe the first machine teachers were Frank Rosenblatt and his colleagues at the Cornell Aeronautical Laboratory in the 1960s. They had show and

label image examples in a trial and error manner in order to train their perceptron mark 1, a proof-of-concept machine capable of learning visual patterns[4].

[4] Researchers in the 1960s teaching a machine to recognize gender from face images: https://youtu.be/cNxadbrN_aI. Source: BBC

IMT, as defined by Simard et al. [Simard et al., 2017] and Ramos et al. [Ramos et al., 2020], is a recent and promising approach that offers a novel and concrete vision of how interactive ML systems should be designed from both an HCI and a software engineering perspective. The IMT approach is presented in more detail in the next chapter, in section 2.2.

IMT could empower millions of people with data automation and processing without computer science expertise and lead to more understandable AI systems that are easier to maintain. In doing so, IMT could simultaneously meet the demand for data automation in specialized areas (medicine, law, science etc.) and the demand for more transparent AI systems.

1.2 Research approach

This thesis takes an empirical approach rather than software engineering by focusing on how people develop teaching strategies and understanding of interactive ML systems through the act of teaching. Investigating human behaviors and understanding when teaching a machine can shed light on what afford ML system and therefore, inform the design of more teachable systems.

The teaching scenario considered in this thesis involves users in the training of image classifier with fast iterations between training and evaluation. It is quite a minimal IMT scenario considering more advanced workflow involving semantic feature decomposition or explanations [Ramos et al., 2020]. However, in the scenario considered, users can create their own training data on the fly, either by sketching or with a webcam, which allows them to create teaching curriculum and strategies, to challenge or steer the machine in the direction they decide. They receive instant feedback on the system's learning status, which influences their choices and understanding. This thesis investigates how people perceive and use ML uncertainty in this teaching scenario.

This interactive approach to ML has already been studied with simple models in IML research [Fails and Olsen, 2003] which training could

be seen as a calibration rather than teaching. Users' behavior and understanding remain to be studied when teaching more expressive models, able to learn abstract representations with complex data. For this reason, this thesis takes a particular interest in artificial neural networks as machine learners i.e. the underlying ML algorithm of our IML systems.

Finally, IMT can be a tool for ML education. Involving novices in training an ML system and exploring its limitations can develop their literacy about ML. Scientific popularization collaborations with the association Traces influenced this thesis, which discusses the pedagogical benefits of engaging novices with the expressive capabilities of modern ML (deep learning) and conveying rich concepts through data.

1.3 Research methods

Both IML and IMT processes are inherently co-adaptive, driven by the user, but both the model and the user evolve together when exchanging information [Dudley and Kristensson, 2018]. Changing the systems' ML model or feedback can disrupt the co-adaptive process between human "teachers" and learning machines. For this reason, human-centered methodologies that captures people's experiences at the moment they teach the system is one of the relevant approach to understand interactive machine teaching as a phenomenon. More generally, the methodology in this thesis triangulates between observation, theory, and software design, following the framework of Mackay and Fayard [Mackay and Fayard, 1997] for Human-Computer Interaction (HCI) research. Observations, theory, and software design "constantly evolve and influence or change models at the theoretical level and observations at the empirical level." Figure 1.2 summarizes the triangulation of this research, illustrating the sequence of empirical, design and theoretical contributions that informed each others along the research process. The course of this research was also influenced by external events such as the covid-19 crisis which required the conduct of remote user studies, and invitations to organize science popularization workshops with the Traces association, which shaped the design of IMT systems for the general public.

The following subsections briefly detail the empirical and analytical methodologies used, which might be new to readers outside of HCI.

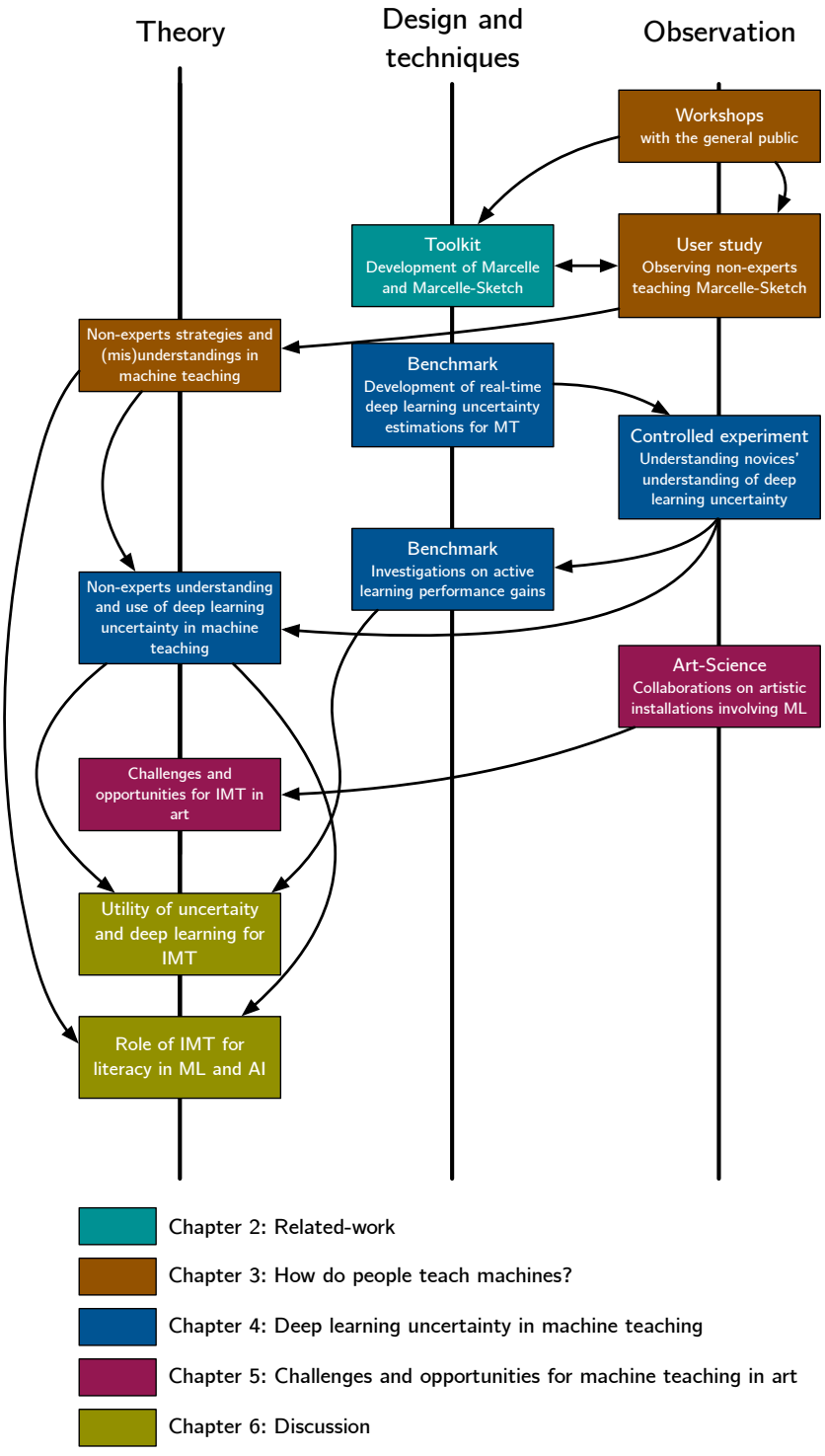


Figure 1.2. Methods' triangulation in this thesis. This research mixes qualitative methods such as structured observations of non-experts users teaching interactive ML systems with the design of teachable systems (using Marcelle) and evaluation of techniques for estimating better uncertainties in IML systems. All the "bricks" are connected with a common theoretical foundation from HCI and ML.

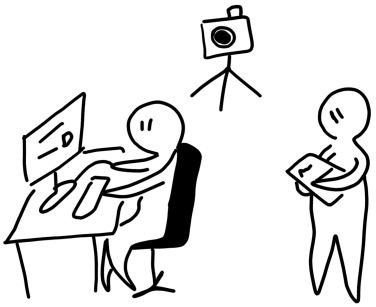
Participatory workshops

Participatory workshops involve a community of people in an exchange with researchers. They mainly help researchers decide what research would be helpful regarding communities of practices [Northway et al., 2014]. In HCI, participatory workshops often involve future end-users in creating artifacts and knowledge to support the design of technologies beneficial to them. The core elements of the workshop are participants, facilitators, information systems, tasks, and places [Numa et al., 2008]. In collaboration with the association *Traces*[5], my research collaborators and I conducted several participatory workshops with various kinds of audiences during science popularization events on a festival, online, and in high schools. These workshops are briefly presented in section 3.2 and influenced this thesis' research directions and methodologies.

[5] Traces is a think-and-do, nonprofit group interested in science, its communication, and its relationship with society: https://www.groupe-traces.fr/

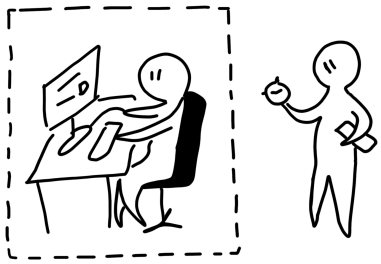
Structured observations

Structured observation [Garcia et al., 2014, Mackay, 2014] is a form of quasi-experiment [Cook et al., 1979] in which researchers employ experiment design principles to compare tasks and gather observational data to increase their understanding of a problem. This is a generative methodology in which participants perform realistic tasks in real-world settings. This methodology does not verify a hypothesis but allows researchers to identify novel user behavior while enhancing ecological validity to respond to nuanced qualitative research questions that cannot be quantified yet, or at all. Structured observation allows researchers to explore promising issues, enhance the discovery of new ideas, generate design implications, and gain insight into using technology in real-world settings. The study presented in section 3.4 is informed by this approach. The study uses a think-aloud protocol as well as both qualitative and quantitative analysis to understand users' strategies and (mis)understanding when teaching a machine.



Controlled experiments

Controlled experiments seek to establish cause-effect relationships. They usually try to observe correlations between independent variables i.e. study conditions, and dependent variables i.e. quantitative measures of a phenomenon in order to test the validity of hypotheses. I used a controlled experiment mixed with structured observations to investigate users' perception of ML uncertainty in a machine teaching task. The experiment is described in section 4.4 and studies the type of uncertainty shown as an independent variable. We complemented this controlled and quantitative approach with a qualitative analysis of



think-aloud verbalizations and semi-structured interviews.

Semi-structured interviews

Semi-structured interviews are a particular form of interview study where an interviewer explores a set of pre-defined themes with the interviewee. For technology design, semi-structured interviews can be oriented towards incidents or objects.

The *critical incident technique* collects "direct observations of human behavior in such a way as to facilitate their potential usefulness in solving practical problems" [Flanagan, 1954]. The technique relies on reporting critical incidents, which are events associated with a system failure and with a special significance for the user. The method focuses on reporting facts rather than interpretations, opinions, or general impressions regarding systems failures.

The *critical object techniques* described by Mackay [Mackay, 2002] is a variation of Flanagan's critical incident technique, which encourages interviewees to recall and reflect on past experiences by sharing and describing artifacts they created. For instance, interviewees could explain how they organize files by describing and reflecting on their actual virtual desktop on their personal computer.

This research performed semi-structured interviews with participants after the two laboratory studies presented in Chapters 3.4 and 4.4. The goal of these interviews was to reflect on their experience as machine teachers. The semi-structured interviews conducted are similar to the critical incident technique, except that we were also interested in interpretations of the system's behavior and how they changed throughout the task. These interpretations convey beliefs and priors on novice's mental model that influences users' teaching choices. Understanding these influences is a central challenge of this research.

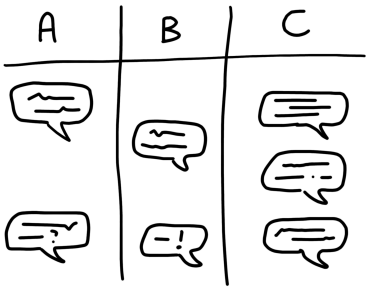
Think-aloud protocol

Think-aloud protocols involve participants thinking aloud as they are performing a task to make thought processes as explicit as possible. A think-aloud protocol provides researchers insights into the participant's cognitive processes rather than only their state of mind after the task. We usually distinguish talk-aloud and think-aloud protocols. Talk-aloud only encourages participants to describe their actions, such as what they are looking at or interacting with. Think-aloud encourages participants to describe their thoughts, interpretations, and

feelings. The downside of think-aloud or talk-aloud protocols is the high cognitive load associated.

I used a think-aloud protocol in both the structured observation study and controlled experiment presented in sections 3.4 and 4.4. Participants had to speak their thoughts while teaching the system with their own data. This method plays an important role in this thesis methodology since it represents a direct technique to approach users' evolution of their mental representations of the machine learner, which is one of the research goals of this work.

Thematic analysis



Thematic analysis is a well-established analysis method in psychology and HCI research to analyze qualitative data [Braun and Clarke, 2006]. Like grounded theory [Strauss and Corbin, 1998, Walker and Myrick, 2006], it involves an iterative process in which the researchers associate codes to transcribed verbalizations, and craft themes to characterize the observed phenomena. However, it differs from grounded theory because the objective is not to generate a theory of behavior but address research questions. I used thematic analysis to interpret interviews and think-aloud verbalizations in both of the laboratory studies presented in sections 3.4 and 4.4. The analysis procedure and themes are further explained in both experimental protocol sections.

1.4 Contributions and statement

In this section, I present and discuss the empirical, technological, methodological and theoretical contributions developed in this dissertation.

Empirical contributions

From an empirical point of view, I conducted two user studies to understand how people teach machines and apprehend uncertainty. This thesis shows that:

- Non-expert machine teachers engage with different teaching strategies that shape both model performance and their understanding of the ML model behavior. When free to explore, non-expert users actively investigate features to test and refine their hypothesis about the model's learning behavior.

- Non-experts' ability to predict the model's behavior depends on the characteristics of their teaching curricula rather than uncertainty feedback. However, they can perceive the difference between aleatoric and epistemic uncertainty in specific situations.

- In our context of incremental machine teaching, a simulated AL curriculum exhibits no performance gain over random data selection. Users' teaching curricula lead to better performances given similar number of examples.

Technological contributions

- This research contributed to the design of Marcelle[6], a toolkit for composing IML workflows and interfaces. This thesis discusses how Marcelle can scale to IMT systems.

- We designed a shareable sketch-based IML system for studying non-experts users in a machine teaching task[7].

- We designed and evaluated a ML pipeline to calculate real-time deep learning uncertainty estimates (aleatoric and epistemic) with transfer learning. These uncertainty estimates were included in IML system to detect ambiguous and novel images within a machine teaching workflow [8].

[6] https://marcelle.dev

[7] https://marcelle-sketch-v2.netlify.app/

[8] Demonstration video for IUI 2022: https://youtu.be/e-2XLLVxjlg

Methodological contributions

- This thesis demonstrates how machine teaching can be used as a method to investigate peoples' understanding and appropriation of ML-based systems.

Theoretical contributions

- This thesis highlights machine teachers strategies and (mis)understandings when teaching an IML system with data they create themselves and toward ML uncertainty.

- It also discusses the advantages and drawbacks of deep learning in teachable systems as well as the utility of calculating two types of uncertainty.

1.5 Thesis overview

This chapter introduced the goals, methodologies, and contributions of this research. I now present the content and organization of the following chapters.

Chapter 2 defines interactive machine learning (IML) along with existing research focusing on specific groups of users. It then presents the goals and characteristics of the Interactive Machine Teaching (IMT) field. Finally, it introduces Marcelle, a toolkit for composing IML workflows and interfaces, which is tightly coupled with the research presented in this thesis and could support the design of IMT systems.

Chapters 3 and 4 present two user studies, structured observations, and a controlled experiment that let novices teach an IML systems with image data they curated themselves by either sketching or using a webcam. These studies are the core of this thesis contributions and focus on what users understand from ML algorithm behavior and what strategy they may use to "make it work." Although both studies have a similar study design, the controlled experiment in Chapter 4 focuses on users' understanding and use of two types of Deep Learning uncertainty for "teaching" IML systems.

Finally, chapter 5 takes a reflective approach to two contrasted artistic collaborations involving ML algorithms. I discuss the challenges and opportunities for applying interactive machine teaching applied to art through the obstacles encountered.

The insights from the two user studies are discussed in chapter 6. I argue that people develop different teaching strategies that rely on their priors and on the systems' feedback. Their teaching strategies structure the composition of their data (sequencing, number, and variability) and affect their ability to understand and predict the algorithm behavior. I discuss the utility of uncertainty, active learning, and deep learning in IMT. Finally, I suggest that IMT systems could be designed as a tool to support peoples' literacy about ML and AI. A conclusion of this work and future research directions are provided in chapter 7.

I provide three appendices. The first one provides the reader with a brief guide to transfer learning that I believe is relevant to building expressive yet effective IML systems with deep neural network architectures. Intended for non-experts in ML, this appendix is a gateway for transfer learning and gives pointers to essential papers and contri-

butions. The second appendix provides an overview of possible data acquisition scenarios in active learning. Finally, the last appendix talks about the aesthetics conveyed by mode-covering or mode-seeking generative deep learning models.

1.6 Publications and collaborations

Some of the content of this thesis produced publications in international conferences.

The user study in chapter 3 appears in:

> Sanchez, T., Caramiaux, B., Françoise, J., Bevilacqua, F. and Mackay, W.E., 2021. How do People Train a Machine? Strategies and (Mis) Understandings. Proceedings of the ACM on Human-Computer Interaction, 5(CSCW1), pp.1-26.[9]

[9] Presentation video for CSCW 2021: https://youtu.be/x_fNhZP2mBQ

The controlled experiment in chapter 4 appears in:

> Sanchez, T., Caramiaux, B., Thiel, P. and Mackay, W.E., 2022, March. Deep Learning Uncertainty in Machine Teaching. In 27th International Conference on Intelligent User Interfaces (pp. 173-190). [10]

[10] Presentation video for IUI 2022: https://youtu.be/H1S24WSD4OY.

The description of the *Marcelle* toolkit implementation appears in:

> Françoise, J., Caramiaux, B. and Sanchez, T., 2021, October. Marcelle: Composing Interactive Machine Learning Workflows and Interfaces. In The 34th Annual ACM Symposium on User Interface Software and Technology (pp. 39-53).[11]

[11] Presentation video for UIST 2021: https://youtu.be/gkMnUl2OZ-Y

Chapter 2

Background and related work

This chapter first defines interactive machine learning (IML) and existing research that focus on existing practices. It then presents interactive machine teaching (IMT) and its specificities regarding IML. Finally, I present Marcelle*, a toolkit for composing IML workflows and interfaces, which is tightly coupled with the research presented in this thesis and could support the design of IMT systems.*

2.1 Interactive Machine Learning

The field of *Interactive Machine Learning* (IML) lies within the Human-Computer Interaction (HCI) field. It focuses on making the process of building ML models interactive, programming-free, and accessible to a broad range of users [Dudley and Kristensson, 2018]. IML systems can also support the well-established working activities of ML practitioners as well as open the door of ML technologies to new users and practices. In the IML workflow, users drive the training and refinement of an ML model through various interactions: providing examples [Amershi et al., 2011], choosing and refining features [Kulesza et al., 2014, Suh et al., 2019], and selecting high-level parameters, among others. In return, the machine learner can provide performance feedback [Fiebrink et al., 2011], visualizations of errors [Amershi et al., 2015], or guidance [Cakmak and Thomaz, 2012a] to convey what it has learned. A desired characteristic of IML is to create shorter iteration cycles between the different activities mentioned above [Amershi et al., Dudley and Kristensson, 2018]. Users should be able to edit the data, train the model and evaluate its outcomes more fluently and without expertise compared with existing ML programming tools. Meaningful interactions and workflows are likely to improve user trust and understanding of the resulting model [Stumpf et al., 2009b].

Dudley and Kristensson [Dudley and Kristensson, 2018] provide an extensive review of IML research and systems, structured by the type of input data on which the system learns, which strongly conditions the type of interactions and ML algorithm used. The authors propose a general workflow encompassing existing IML systems. The IML workflow can be decomposed into various activities illustrated in Figure 2.1.

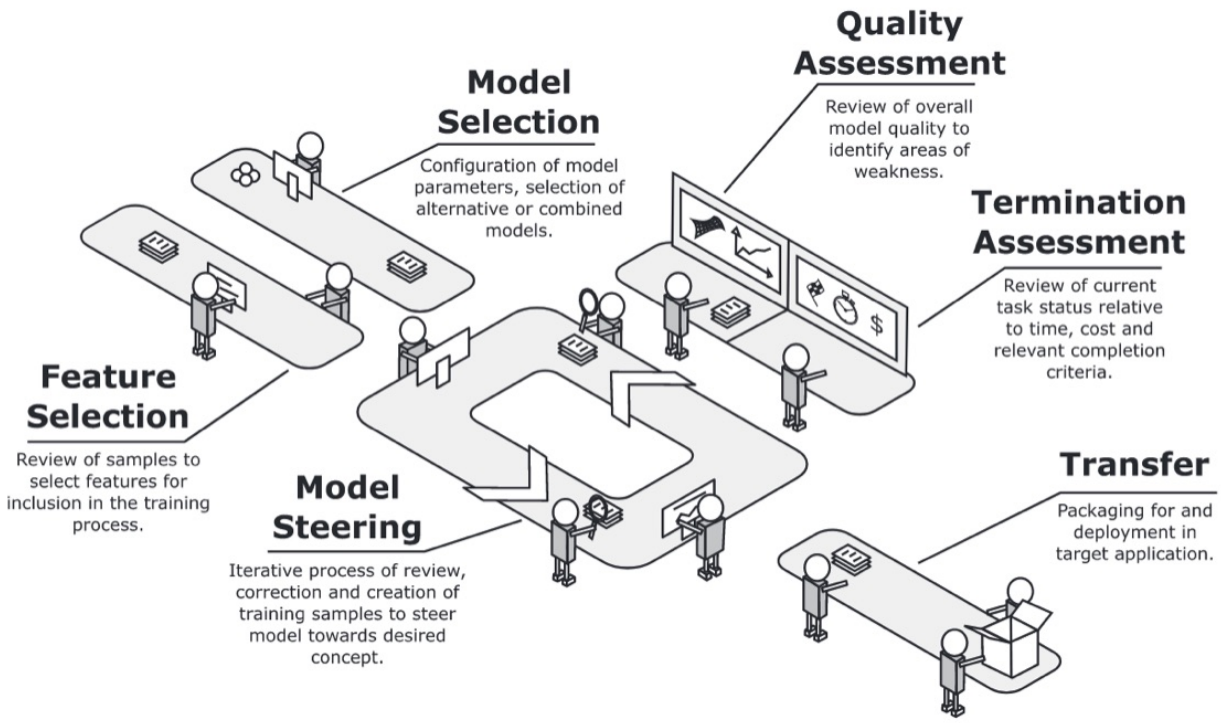


Figure 2.1. The IML workflow, as a behavioural breakdown into distinct user activities. Source: Dudley and Kristensson [Dudley and Kristensson, 2018].

Several guidelines for the design of IML systems have been proposed from empirical studies involving users interacting with IML systems [Amershi et al., 2019]. From their literature review, Dudley and Kristensson [Dudley and Kristensson, 2018] propose six design principles: (1) make task goals and constraints explicit, (2) support user understanding of model uncertainty and confidence, (3) capture intent rather than input, (4) provide effective data representations, (5) exploit interactivity and promote rich interactions, (6) engage the user.

In practice, new interaction techniques and visualization have often focused on one or a subset of the activities presented in Figure 2.1, such as data iteration [Hohman et al., 2020] or quality assessment [Fiebrink et al., 2011]. Similarly, IML integrates into various existing expert prac-

tices, either in computer science or others, resulting in original iterative workflows that push the boundaries of these activities illustrated in figure 2.1. In the following subsection, I present some examples of IML research that apply to specialized users.

ML experts

ML experts gather professions that traditionally deal with the production of ML models: ML researchers, ML engineers, and data scientists. The IML literature regarding ML practitioners mainly focuses on improving a part of the ML workflow e.g. improving performance metrics, iteration on data, and debugging models through more transparent feedback.

For instance, Hohman et al. [Hohman et al., 2020] showed that ML practitioners improve model performance primarily by iterating on training data (i.e. collecting new data, adding labels) rather than iterating on the model (i.e. architecture and hyperparameters). Furthermore, if versioning is a widespread tool to iterate on code, there may not be any equivalent for training sets. The authors designed Chameleon, depicted in Figure 2.2, a collection of interactive visualizations for training set versioning in an ML classification task. ML practitioners can navigate timelines allowing comparisons between different versions of data sets and inspect changes in features and performance measures after retrainig the model.

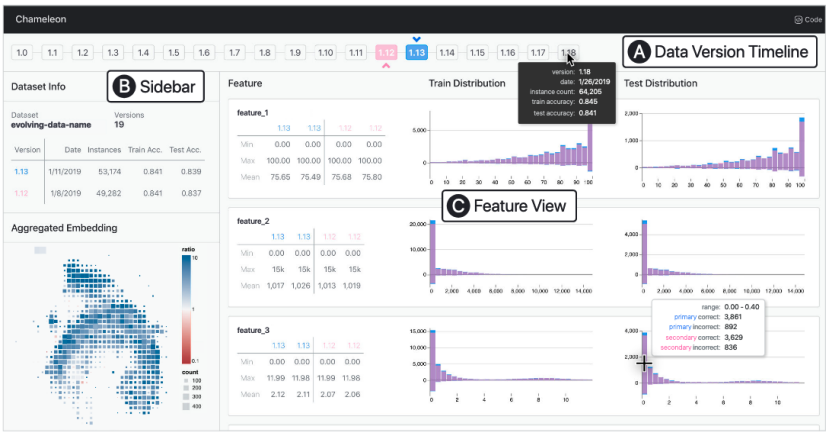


Figure 2.2. Chameleon, a collection of interactive visualizations for ML versioning in a classification task [Hohman et al., 2020]

For the evaluation phase, ModelTracker, depicted in Figure 2.3, is an interactive visualization for examining and debugging binary classifiers at the level of individual examples [Amershi et al., 2015]. Users can inspect and correct mislabeled instances and inadequate features

throughout the model building.

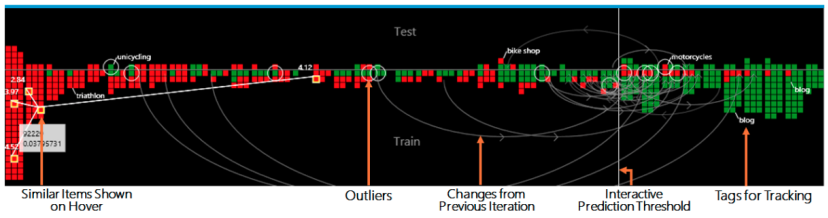


Figure 2.3. ModelTracker: an interactive visualization for example-level performance examination and debugging of binary classification [Amershi et al., 2015]. A classifier is well trained when all red dots are separated from the green dots on the horizontal scale representing the prediction probability. Arrows represent classification changes at the instance-level from a model state to another.

A typical task for ML expert users is to compare several trained models. EnsembleMatrix, represented in Figure 2.4, is an interactive visualization enabling practitioners to explore an ensemble of models and build combinations of models to improve performance [Talbot et al., 2009]. Users can inspect the confusion matrix of several models and the confusion matrix of the combined classifier.

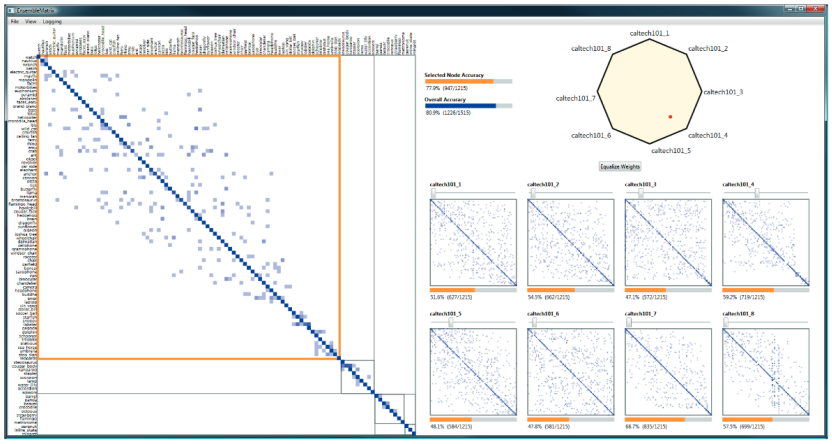


Figure 2.4. EnsembleMatrix: interactive visualization of confusion matrices to explore and build combinations of models for image classification [Talbot et al., 2009]

The systems presented above require expertise to comprehend and operate them. They focus on a single activity of the ML workflow. This thesis does not focus on ML experts, but it is worth mentioning existing work for ML practitioners that constitute an important share of the IML research and could inspire the design of IML dedicated to non-expert users.

Another important line of IML research is to empower non-expert users (in ML or computer science) with the predictive and automation capabilities of ML. A major challenge is understanding how specific or general the design of IML systems should be. Are there principles of IML interaction that apply to all types of users? It appears that this

is also what the IMT domain seeks to accomplish, as presented in section 2.2. Regarding non-expert users, Yang et al. [Yang et al., 2018b] conducted interviews with different types of non-experts to understand the opportunities and breakdowns when novices (in ML) build ML solutions for themselves in real life. In the following sections, we present the research and ML systems that have been considered for non-experts, including various types of subject-matter experts who have specific goals and existing practices of technology: designers, developers, creatives and artists, scientists, and the general public.

Designers and developers

Designers and developers both design interactions for future end-users. IML can help them to improve user experience with data-driven interaction design. For example, ML can help game developers create gameplays that exploit players' gestures and poses through motion sensors or video frames from a webcam. These types of interactions are difficult to build with explicit programming. However, ML is a tedious design material for HCI practitioners [Yang et al., 2018a]. Dove et al. [Dove et al., 2017] identified four design challenges for interaction designers: the difficulty (1) to consider the interplay between the ML statistical intelligence and common-sense human intelligence, (2) to apply ML in less obvious ways, (3) to represent ML dependency on data in early prototypes and (4) to envision the ethical considerations of ML.

Fails et al. [Fails and Olsen, 2003] introduce Crayons, an IML proof-of-concept intended to provide an efficient method for developers and designers to create Perceptual interactions i.e. interactions that rely on high-dimensional data and are difficult to implement explicitly. Crayons is a camera-based IML system for automatic image segmentation. Users can paint on the image directly to identify foreground or background areas. The model is retrained after each annotation. It is historically one of the seminal papers introducing IML.

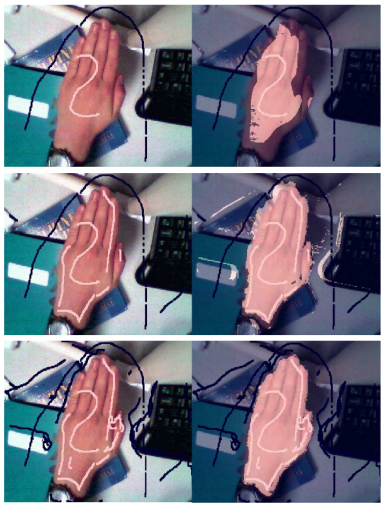


Figure 2.5: The iterative training process of Crayons, a camera-based interactive segmentation system [Fails and Olsen, 2003]

IML workflows and tools have also been developed for developers, in particular, game developers [Diaz et al., Xie et al., 2019, Bernardo et al., 2017]. InteractML [Diaz et al.] is a visual programming extension for Unity3D dedicated to game developers and designers willing to explore perceptual interactions for new gameplay mechanics. InteractML enables to can create ML models by joining nodes together and visualizing in real-time the data from a Unity scene in the graph. In this situation, IML systems are analogous to no-code development platforms (NCDPs) that allow developers to create application soft-

ware through graphical user interfaces and configuration instead of traditional computer programming. InteractML does require basic concepts of ML, such as the difference between a classification and a regression, as well as being familiar with the Unity3D environment. This work demonstrates how IML can support the adoption of ML technologies within existing practices and tools (e.g. a popular game engine) in order to be adopted by a specific community of users.

Creatives and artists

An IML research community has also developed around performing arts and musical applications. For instance, the IML workflow was used to build musical instruments and movement-to-sound regression mapping for live performances [Françoise and Bevilacqua, 2160]. As an example of a system, the Wekinator [Fiebrink et al., 2009] is a standalone software using ML algorithms to map arbitrary inputs and outputs in real-time. It uses an analogy of input and output signals and the OSC format, which are popular concepts and formats of the audio community. The arbitrary inputs and outputs mappings allow for broader uses than in music and performing art. More specialized libraries integrate ML algorithms into the popular digital sound processing (DSP) platform and graphical programming language Cycling'74 Max: the ml.lib [Bullock and Momeni, 2015] include general-purpose models whereas XMM [Françoise et al., 2014] focus on specialized and probabilistic models.

These IML tools offer the possibility to quickly design and play digital instruments with various input modalities [Fiebrink and Caramiaux, 2018]. Morris et al. [Morris et al., 2012] conducted workshops with music students to design and play digital instruments created with IML tools. Their results highlight the role that IML can play in pedagogy: not only did music students discover the possibilities of ML, but they were also engaged in high-level creativity and social interaction regardless of sensorimotor or theoretical skills. In other words, IML offers personalized controls to express musical intentions that exempt students from mastering a musical instrument or music theory.

IML fosters inclusive design techniques to define gesture-sound mapping *by demonstration* or *through listening* in which users create the mapping by performing on a sound they hear [Caramiaux et al., 2013, 2015, Françoise and Bevilacqua, 2160]. Once trained, users can interact with the sound, and perform in real-time, as illustrated in Figure 2.6 taken from [Françoise and Bevilacqua, 2160]. This design and performing process can be carried out by a wide range of users with different

motor skills: professional artists, children, or disabled people [Katan et al., 2015, Scurto and Fiebrink, 2016]. The training phase in this research also involves the performer in a tight interaction loop but could be seen more like a calibration than a teaching process.

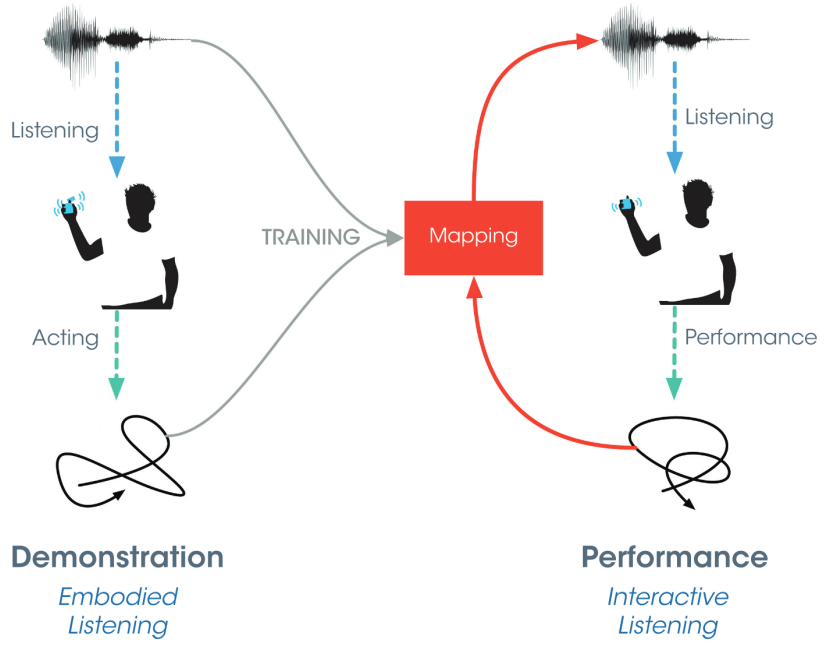


Figure 2.6. Overview of the workflow of *Mapping through Interaction* taken from Francoise and colleagues [Françoise and Bevilacqua, 2160]. Blue and green dashed arrows respectively represent listening and moving. In Demonstration (top), the user performs a movement while listening. Both movement and sound are used to learn an interaction model (mapping). In performance (bottom), the user's movements continuously control the sound synthesis with the learned mapping. Source: [Françoise and Bevilacqua, 2160] with the author's consent.

The adoption of ML techniques by artists and creatives also challenges assessment criteria. Creatives might have subjective criteria for quality assessment of machine behavior. As an example in the musical domain, Fiebrink et al. [Fiebrink et al., 2011] found that non-expert users also use qualitative measures such as unexpectedness and real-time evaluation to reflect on the data curated in addition to simple performance metrics such as accuracy and cost. Furthermore, the IML workflows can foster exploration and discovery. Co-Explorer is a parametric synthesizer using deep reinforcement learning. Users can give positive or negative rewards while the sound is being generated [Scurto et al., 2021]. Co-Explorer offers a new co-exploratory workflow for sound designers in which the IML process may become more important than the learned model itself.

Aside from artistic performance, generative or image processing ML models have appealed to many artists or graphic designers. These models are more significant technical obstacles than the simpler regression models mentioned earlier because they usually involve deep and convolutional architectures, implying heavier computational re-

sources. If platforms exist to popularize these technologies (e.g. RunawayML), they generally do not involve end-users in a tight interaction loop, enabling them to train models incrementally.

Scientists

Research in IML aims to support other scientific practitioners in the medical field, biology, or robotics. In the medical, biomedical, and biology fields, IML systems mainly support decision making or automate laborious tasks that comprise small, complex datasets with rare occurrences [Holzinger, 2016]. Ilastik [Berg et al.] is an IML system that applies to (bio)image analysis for image segmentation, object classification, counting and tracking. Practitioners can interactively update the model with sparse annotations on few images and iterations on the system predictions, in the same fashion as Crayons[1] [Fails and Olsen, 2003]. Similar tools were developed to support specific image-analysis tasks in cardiovascular research [Razeghi et al., 2020] and radio-therapy [Smith et al., 2022].

[1] Ilastik demo: https://youtu.be/5N0XYW9gRZY?t=62

IML can also foster knowledge discovery, and data mining in the scientific domain [Wallace et al., 2012, Holzinger and Jurisica, 2014, Cai et al.]. Abstrackr [Wallace et al., 2012] is a collaborative IML tool intended to support researchers to screen research articles. The system shows article abstracts to end-users, who can tag relevant or irrelevant keywords to refine an SVM model. Users can also tag a document to be relevant or not, which updates the model. Abstrackr samples new abstracts using Active Learning, which aims at selecting the most uncertain document i.e. lying on the decision boundary between relevance and irrelevance. The IML workflow also leverages end-users trust because they are involved in the training process and develop more accurate mental model of the system capabilities than if they used a ML system "out-of-the-box" [Cai et al., Guo et al., 2022].

IML workflow also applies to robotics in order for scientists or users to demonstrate a particular behavior to a robot [Lee, 2017]. Users can directly manipulate a robotic arm [Ravichandar et al., 2020] (called kinesthetic teaching), give rewards or penalties when the robot tries a movement by itself [Chernova and L. Thomaz, 2014, Thomaz and Hoffman], demonstrate segments of a movement, etc. Consequently, the Human-Robot Interaction field also studied teaching modalities enabling humans to convey concepts to robots [Thomaz and Breazeal], as well as robots to be proactive and engage with the users with queries [Cakmak and Thomaz, 2012a, Chao et al., Racca et al.]. For instance, Cakmak et al. [Cakmak and Thomaz, 2012b] found that human teachers are

often sub-optimal when conveying a concept among a set of possibilities and active learning help human teachers to converge faster to the concept despite being sometimes counter-intuitive. Thomaz et al. [Thomaz and Breazeal] found that participants tend to give more positive than negative rewards in reinforcement learning settings. These results are often transferable to HCI considering similar task complexity, even though the fact a robot embodies the learning system might affect users' perceptions and interactions.

Figure 2.7. A human teaches concepts with colored paper patterns to the robot Simon. Figure taken from [Cakmak and Thomaz, 2012b]

The general public

IML offers the general public tools to create and control personalized AI systems [Amershi et al., Gillies et al., 2016, Wolf et al., 2018]. The general public might also have many unlabeled data they might benefit from labeling to filter information. For instance, ElluciDebug is an email-like text management and classification tool [Kulesza et al., 2015] develop as a proof of concept to illustrate the notion of *explanatory debugging*. The system comprises several folders i.e. categories in which emails are classified. The system provides explanations along with predictions. End-users can interact and change the explanations if inadequate, giving both instance or feature-based corrections. IML can also support the general public to browse information from large online databases. For instance, CueFlik is an interactive Web image search tool [Fogarty et al., 2008]. Rather than search keywords, users can add positive and negative image examples to update the ranking of a Web image base [Amershi et al., 2011]. The CueFlik interface is illustrated on figure 2.8.

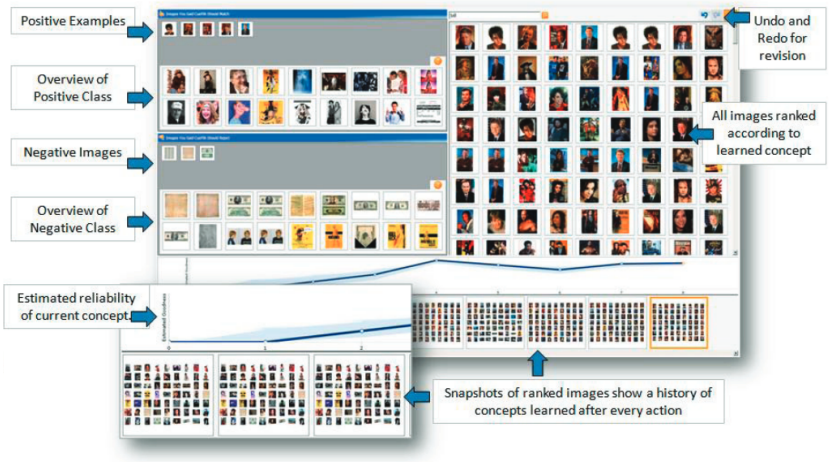


Figure 2.8. CueFlik: a web image search application using positive and negative image examples to train a searching model interactively [Fogarty et al., 2008]

Aside from empowering end-users in their interaction with technology, an important thread of IML research aims at designing IML systems for ML education i.e. how interaction with an IML system can make people learn about ML. Few research works engage children with IML to develop literacy with ML technology. Agassi et al. [Agassi et al., 2019] designed a gesture recognition IML component in Scratch, a visual programming language dedicated to children [Resnick et al., 2009]. Along with the Scratch modules, they designed a physical device with embedded accelerometers. The authors aimed at encouraging children to include gesture recognition in their Scratch project, allowing them to collect gesture data by themselves and train the model through trials and error. The authors argue that fostering an early understanding of ML processes through the game and direct manipulation can help children later understand more complex ML systems. Dwivedi et al. [Dwivedi et al.] reports outcomes from workshops with children in which they used an IML system to classify origamis. The authors argue that IML could help children develop creativity and comfort with ML and AI. In another work, Hitron et al. [Hitron et al., 2019] showed that teaching a gesture-based recognition system fosters children's understanding of machine learning mechanisms, and this knowledge can be transferred to applications from everyday life. With older students, a similar approach has been explored in sports with young athletes to foster introspection on athletic movements [Zimmermann-Niefield et al., 2019a].

Teachable Machine is an online application to create, customize and export ML classifier [Carney et al., 2020a]. The authors' primary goal was to help students and teachers learn, teach and explore ML concepts through interaction. They have recently tried to fill the

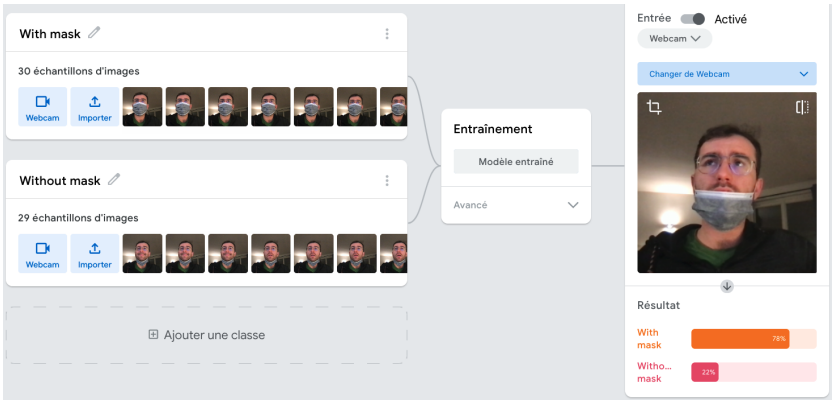


Figure 2.9. Image classification from Teachable Machine. Users can curate images with their webcam, train a classifier and perform real-time prediction for model assessment [Carney et al., 2020a]. The different activities in the IML workflow are represented by boxes connected with cords.

gap between ML and the makers' community by enabling model exports for micro-controllers. This tool is largely used in educational ressources [aic, 2019].

Summary

The IML research emphasizes empowering people, either ML practitioners (out of the scope of this thesis), subject-matter experts, or the general public, with automation, processing and predicting capabilities. Subject-matter experts can benefit from IML tools in diverse ways. The IML research demonstrated that developers and designers could create new data-driven interactions without explicit programming. Creatives and performers can build embodied interactions with sound or images that foster exploration. Scientists and medical doctors can benefit from IML to automate tasks, discover knowledge, or support decision-making. In several examples, the IML workflow was embedded in existing practices and tools, which has the benefit of engaging users with familiar vocabularies and interactions [Ramos et al., 2020].

2.2 Interactive Machine Teaching

This section describes *Interactive Machine Teaching* (IMT), as defined by Simard et al. [Simard et al., 2017] and Ramos et al. [Ramos et al., 2020]. In particular, it highlights the specificities of IMT, which is belongs to the more general IML research.

Specificities of IMT research

IMT can be seen as a subfield of IML with more specific characteristics: (1) a specific goal (*model building* rather than *task completion*), (2) the specific role users are involved in (teachers), and (3) the specific types of interactions that leverage people abilities to convey knowledge [Ramos et al., 2020]. The two subsections detail these characteristics. The third point includes design considerations, and will be discussed in a separate section 2.2.

IMT Goals

First, IMT aims to **leverage the barrier for subject-matter experts to build ML models**. IML, on the other hand, applies to a wider variety of users and is often task-specific [Amershi et al., 2015, Hohman et al., 2020]. IML integrates into existing practices and tasks as seen in section 2.1, including those of ML experts. ML models, as objects, are not necessarily the goal but also a means to accomplish a task [Scurto et al., 2021]. The goal of IMT is to enable users to build ML models to perform future tasks that go beyond a one-time-only specific task. It appears that most situations and systems can pursue both objectives like in segmentation tools for biomedical research introduced above in subsection 2.1. In this case, the goal is to facilitate and automatize laborious tasks (biomedical image segmentation) and create ML models that can be shared with collaborators. In analogy with Grudin [Grudin, 2005], I believe IMT strive for a more discretionary hands-on use of ML i.e. people and workers should possess the ability to build ML models if they wish [2].

[2] On the opposite, micro-workers interaction with ML could be qualified as a mandatory hands-on use i.e. people have no choice than performing the proposed data labeling task and do not have control on the trained model.

The role of teacher

Second, Ramos et al. [Ramos et al., 2020] emphasizes the role embodied by users. In a socio-technical context, a role can be seen as a set of connected behaviors and beliefs people conceptualize when interacting with others and technology. As mentioned in the introduction, only ML experts are involved in the creation of ML models nowadays (which may seem reasonable at first glance). IMT can enhance novices or subject-matter experts in the role of teachers rather than mere annotators and empower them to create specialized ML models for their own needs. Having ML knowledge does not disqualify a person as a machine teacher, but the ML knowledge (model architecture, hyperparameters, etc.) should ideally be out of the scope of the language of Interactive Machine Teaching.

According to Ramos et al. [Ramos et al., 2020], a user acts as a machine teacher if they:

1. **Plan and update the teaching curriculum**: the teacher is free to arrange explanations, the type and timing of evaluations, the task's goal, and choose when and how to fix the model.

2. **Explain knowledge pertinent to the subject domain**: to make the teaching process more efficient, Ramos et al. [Ramos et al., 2020] emphasize the importance of IMT systems to enable users to express rich semantic knowledge. In practice, this translates into the possibility to create, decompose or merge features or tasks, and provide information beyond labeling.

3. **Review the learner's progress while interacting with the given knowledge**: as in IML, the interaction is a dialog in which the machine learner can provide meaningful feedback on its learning status. This feedback (or explanations) influences the machine teacher's next action, which makes IML (and IMT) a co-adaptive process by nature [Dudley and Kristensson, 2018].

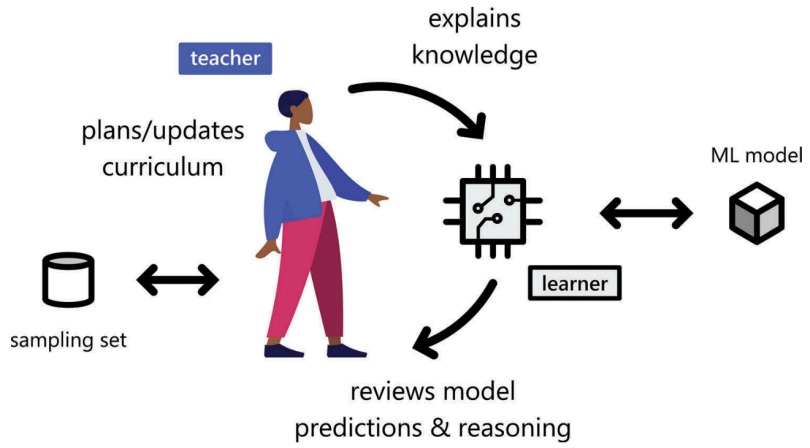


Figure 2.10. The IMT loop. During an interactive machine teaching session, a human teacher and a machine learner communicate iteratively through a teaching vocabulary (labels, examples, explanations, etc.). While engaged in this process, teachers plan the curriculum, explain knowledge, and review the learner's state and model's predictions. Source: [Ramos et al., 2020]

Machine teachers intertwine these three activities over time. This co-adaptive interplay between these three desired behaviors of machine teachers is called the IMT loop [Ramos et al., 2020], and is depicted in Figure 2.10. The IML literature also depicts similar representations of this co-adaptive dialog between a user and a learning system [Amershi et al., Dudley and Kristensson, 2018, Fails and Olsen, 2003]. However, the IMT loop emphasizes human-centered behaviors that characterize machine teachers.

Dialog between human teachers and machine learners

IMT should include specific types of interactions that leverage people's abilities to convey knowledge. Ramos et al. [Ramos et al., 2020] identify four inherent capabilities that should serve as material for designing IMT interactions. These four abilities and their descriptions in the scope of IML are listed aside in Figure 2.11.

Judgement: capability to make decisions about the label value of particular example.

Insight: capability to analyze and synthesize relevant knowledge.

Foresight: capability to articulate knowledge solely from prior subject-domain knowledge, without the need to see a particular piece of information.

Sensemaking: capability to interpret the information they see so far and translate into hypothesis regarding ways in which information and concepts relate through higher-level concepts, structures, or relationships.

Figure 2.11: Intrinsic human capabilities involved in teaching. Source: [Ramos et al., 2020]

Teaching beyond labels

To support the inherent human capabilities to teach, **IMT systems should encourage people to produce knowledge beyond labels**. Many IML systems are limited to producing labeled data, but knowledge can be transmitted through many forms. For instance, selecting a set of data without necessarily attributing any labels can already be informative for a learner. Reinforcement learning (RL) enable users with critiques, such as in Co-explorer [Scurto et al., 2021] for sound exploration or in human-robot interaction [Chernova and L. Thomaz, 2014]. Few works explored text prompt as a modality to teach interactive RL models [Krening, 2018, Krening et al., 2017], that might be fueled by a growing literature on deep learning that bridges multiple data modalities [Radford et al., 2021, Baevski et al.]. Several works recommend and investigate the use of semantic features, which are necessary for the model to be interpretable by people [Ramos et al., 2020, Ng et al., 2020]. AnchorViz, for instance, is a data visualization technique for IML in which users can place semantic features on a circular data visualization. The anchors (semantic features) act like magnets on data points i.e. the data points that match the feature are attracted to the anchor. Placing several anchors can help users to separate data on the visualization. IMT argues for enabling machine teachers with the possibility to create, merge, compose or decompose semantic features because people build knowledge on top of other knowledge building blocks. In this direction, Kulesza et al. [Kulesza et al., 2015] demonstrated that structured labeling (i.e. enabling a hierarchy of labels) could help users in being consistent in their labeling and avoid concept evolution. However, they do not tackle features crafting but rather structured annotations at the instance level. Ramos et al. [Ramos et al., 2020] tackled feature decomposition with text documents. The authors envisioned PICL, an IMT system in which machine teachers can compose features extracted from several models they taught to form a schema. For instance, a taught extractor (e.g. detect if a string of characters is an ingredient) can be used to refine a classifier (e.g. decide if a document is a cooking recipe or not). The ingredient extractor itself could be decomposed into other extractors (quantities, fruit, vegetables, etc.).

Designing actionable feedback

IMT should also provide information about the machine learner so that teacher can affect the learning process outcomes [Ramos et al., 2020]. The IML field also shares this goal but much remains to be done, especially considering the rapid expansion of the explainable and intelligible AI (XAI) research field [Gunning et al., 2019]. Research on intelligible AI systems was impulsed by the "right to explanation" in the European General Data Protection Regulation [Goodman and Flaxman, 2017] and focuses on providing explanations in addition to predictions. Explanations are feedback that supplements predictions by answering the question: "why was this prediction made?". Despite ongoing discussions on defining and evaluating interpretability in intelligent systems [Doshi-Velez and Kim, 2017, Lipton, 2018], two main approaches stand out for making models more intelligible [Weld and Bansal, 2019].

First, the model can be intelligible by design, such as linear models, rule-based systems and decision trees [Rudin, 2019], which provide explicit rules that subject-matter experts can use without computation. For example, such model can predict if a patient is diabetic and give explicit decision criteria e.g. a patient is diabetic if fasting blood glucose is above $1.26g/L$ or the rate of glycated haemoglobin (HbA1c) is above $\geq 6.5\%$.

These models can provide algorithmic transparency by design but might rely on heavily engineered features for complex data [Lipton, 2018] and can lead to poor generalization. The second approach aims at computing explanations using the model or an approximation of the model itself [Ribeiro et al., 2016, Zeiler et al.]. This approach can be applied to more complex data such as raw images, in which the explainer module can highlight pixels responsible for a prediction, for instance. This approach is criticized because the explainer i a model itself that can be wrong and contributes to the lack of interpretability of the ensemble. Current research in XAI tries to balance or push the limits of the accuracy-interpretability compromise [Lakkaraju et al., 2016, Caruana et al., 2015, Ustun and Rudin, 2017, Valdes et al.], especially in high-stake domains such as medicine.

In IMT, exchanging explanations rather than labels and predictions is a desired characteristic of IMT systems, which could contribute to developing a new language for the dialog between human teachers and machine learners. Explanations were explored in IML research [Kulesza et al., 2015, Kim et al., 2020, Ghai et al., 2020] but play a

central role in the IMT field.

Designing machine learners' interventions

Being active when learning something is generally a desired characteristic of a student. Students that ask questions are perceived as more engaged and motivated to learn, especially if the question points out a relevant ambiguity or novelty. Besides the expressiveness of the signals mentioned above, machine learner interventions' design is a promising direction to enhance people's abilities to understand the ML model knowledge and provide directions for improving it. Interventions can take multiple forms: advice, guidance, and queries, among others. These interventions can be generated by the learning model itself or decided by the designers of IMT systems.

Among interventions generated by the system, Active Learning (AL) is a scenario in which an ML model is allowed to be "curious" i.e. query unlabeled instances on which it should be trained [Settles, 2010]. An uncertainty measure drives the selection criterion: if the uncertainty is too high on a new example, this external information source is queried to a human annotator [Cohn et al., 1996]. AL techniques differ according to the data acquisition scenario i.e. if unlabeled data are available at once (pool-based scenario), as a sequence in time (stream-based scenario), or generated de novo (Membership query synthesis scenario). These variants are illustrated on appendix B. The AL research also explores situations with noisy annotations i.e. erroneous responses to AL queries [Xu et al., 2017] and query formulation that goes beyond labeling instances [Shivaswamy and Joachims, 2012, Kane et al., 2017]. For instance, users could express preferences by reorganizing a ranking (the query). The model could then learn from this improved ranking, even if the ranking is sub-optimal. Related to AL, Active Class Selection (ACS) [Lomasky et al., 2007] focuses on calculating the most beneficial class in which the one should add data according to the model parameters.

Several works in Human-Robot Interaction (HRI) consider AL in an interactive setting to balance agency between the human teacher and the active machine learner. How to share agency in IMT systems belongs to a broader discussion on the tension between automation and direct manipulation in HCI, addressed by Horvitz [Horvitz, 1999, 2007]. Cakmak et al. [Cakmak et al., 2010] compared four different interaction modalities between a human teacher and a learning robot: (1) a fully supervised learning mode (SL) in which the robot does not ask queries, (2) a fully active learning mode (AL) in which the robot only asks queries, (3) an *"any question"* mode (AQ), in which queries are

only human-triggered, and a mixed-initiative active learning (MI) in which queries can be triggered both by the human and the robot. For each of these interaction modalities, participants had to teach concepts such as "house" or "snowman", using colored papers patterns with various characteristics (shape, size, and color). The authors [Cakmak et al., 2010] show that human-controlled active learning (AQ and MI) achieve performance gain over the fully supervised scenario. The use of active learning was preferable for the perception of the robot's intelligence, ease of teaching, and enjoyability. Overall, the "any question" mode (AQ) was the most preferred. Finally, the authors suggest that the optimal strategy is likely to be user-dependent but insist on the need to balance the control early on in the learning process and avoid uninformative queries that might confuse the human teacher. These results might change in different teaching tasks and situations but support the idea that human teachers should not be mere annotators i.e. machine learner interventions should not take over the control of the training. Ramos et al. [Ramos et al., 2020] use AL as a sampler to pick a new document. In their system PICL, active learning sampling coexists with other techniques (keyword search, random selection, possible errors, and predicted positives) as a way for the machine teacher to parse documents to be labeled by the machine teacher. In this case, AL is closer to the "any question" mode introduced in [Cakmak et al., 2010] and the end-user is free to use its sampling strategy. Other work suggests that class imbalances can compromise the benefits of AL in terms of performance and other sampling heuristics should be applied in this case [Attenberg and Provost, 2010].

Machine learner interventions may not only be a query calculated by the system from the model uncertainty. Wall et al. [Wall et al., 2019] designed guidance using teaching patterns from experienced machine teachers. Their goal was to understand if machine teaching skills could be transferred to novices by reifying teaching patterns from experienced machine teachers. The authors applied this concept in a text document labeling task, and guidances take the form of notifications that trigger at specific moments of the teaching. They found that teaching guidance did not improve the classifier's performance trained by novices, but participants expressed less frustration and mental demand.

Toward an IMT language?

Simard et al. [Simard et al., 2017] insist on the need for a standardized IMT language, disconnected from the theory of ML. Formed as a set of user interaction, this language would not only consider labels as the main object users can iteract with but also features and schemas

i.e. and how several ML models can be combined to solve a problem. Along with design patterns and documentation, an IMT language could enable machine teachers to collaborate more fluently. Machine teachers could read, understand, refine, and maintain taught models with a common language, which could help to scale ML model-building across multiple collaborators.

Socio-technical benefits of having machine teachers

First, Simard et al. [Simard et al., 2017] argue that there is a mismatch between the growing demand for ML systems and the ability of organizations to build them. The diversification of roles within ML may follow the same path as IT has experienced in the past. Computing is somehow decoupled between a science, a technology, or a tool. This decoupling started with the first programming languages that have separated physicality from symbols. Developers no longer need to know the inner working of computers (e.g. CPU architectures) to develop software. Similarly, ML practices could diversify and enable intermediate roles that efficiently build ML models (machine teachers) that would be independent from professions that improve existing models and architectures (ML researchers). If the IMT develops as a community, it could foster a new economic market with a more decentralized production of ML models.

Second, IMT principles such as feature decomposition and model schemas (different sub-models organized in a graph) could be a way to lower the technical debt of ML systems. Technical debt is the implied cost of additional rework caused by choosing an easy solution instead of using a better approach that would take longer. ML systems are known to induce heavy maintenance costs at the system level [Sculley et al., 2014], mostly because the training process is performed at once, with large training sets. ML practitioners might need to retrain the model from scratch if the trained model does not perform as expected. This maintenance is even harder to perform when models are monolithic such as deep neural networks trained in an *end-to-end* fashion[3]. Features decomposition is also meant to facilitate ML system debugging.

[3] The *end-to-end* processing, commonly associated with Deep Learning, tends to discard any process of producing intermediate representations: the models take the "raw" data and output the final labels without human intervention such as feature extraction.

Third, IMT can help to meet the desire for ethical accountability regarding deployed ML systems since IMT envisions a more transparent and explicable model-building process. The IMT language is desired to be modular and easier to version and review. Assisting more people in building their own ML systems can balance the power asymmetry between the owners of massively deployed ML algorithms and citizens or subject-matter experts.

How does this thesis position regarding IML and IMT?

IMT is a recent and promising approach that builds upon IML by considering the design of systems that could support peoples' ability to teach a concept or a behavior to a computing system. This thesis is part of this line of research but takes an empirical methodology and focuses on users without ML knowledge [4].

[4] In the following thesis, novices will refer to novices in ML

More precisely, this thesis studies how novice users develop strategies and understand an IML system during a realistic teaching scenario. Users' understanding mainly refers to **functional mental model**, which focuses on users' ability to know how the system behaves as opposed to **structural mental model**, which focuses on a detailed understanding of how and why a system works [Kulesza et al., 2013]. Prior research in this direction by Ng et al. [Ng et al., 2020] and Sultanum et al. [Sultanum et al., 2020] looked at the behavior of human teachers to teach hypothetical learners through formative studies. This work tries to understand how novices teach and understand learning machines under the lens of a working learning system, which can inform the design of IMT systems accessible to the most.

Although the users studied in this thesis are engaged in a tight interaction loop with the machine learner, the IMT systems used in this work are quite minimal regarding the interaction possibilities offered by explanations and features decomposition presented above. Our teaching scenario fits the basic requirements of IMT according to Ramos et al. [Ramos et al., 2020]: *"the simplest ML model-building process deserving of the interactive machine teaching moniker is an iterative process of (1) selecting and labeling of examples and (2) evaluating student learner performance using selected examples."*. The teaching scenarios considered differs from existing IMT work in that users are free to create their own training data on the fly. This characteristic engages people differently than a situation that provides a large unlabeled dataset from the start. Creating their own data allows users to steer the machine's behavior in the direction they want and voluntarily create an instance that can challenge the system.

The second difference of this thesis is that it considers image data and deep architectures of artificial neural networks. In contrast, seminal articles of IMT focus on large collections of text documents [Ramos et al., 2020]. Deep learning models are not recommended in IMT for their lack of transparency and interpretability. However, users' ability to craft features is more challenging with images than text. Deep Neural Networks (DNN) demonstrate the capacity to learn from com-

plex and disparate examples and self-organize different levels of abstraction across hidden layers. The thesis discusses their benefits and disadvantages in the scope of IMT.

2.3 Marcelle: composing interactive machine learning workflows and interfaces.

This section presents Marcelle, a toolkit that allows for designing and implementing IML and IMT applications. With Marcelle, one can compose ML workflows and on-demand interfaces. During my thesis, I used and contributed to the development of Marcelle. This section first presents the toolkit, which was used in the following chapters.

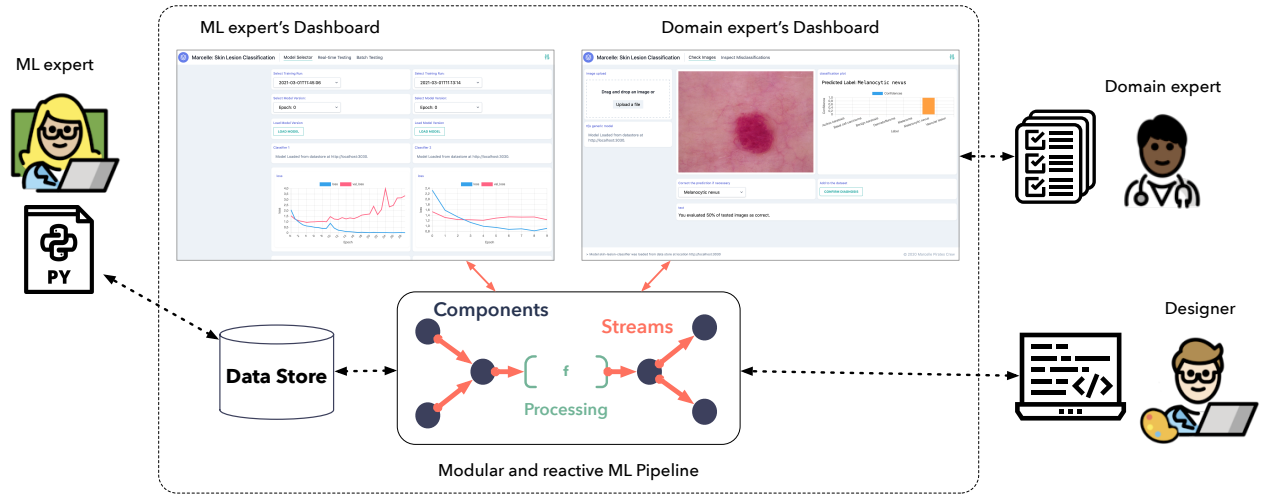


Figure 2.12. Marcelle is a toolkit for IML addressing the composition of custom workflows. It implements a component-based architecture using reactive programming for pipeline specification. Components provide views that can be composed to form custom interfaces. Marcelle's architecture facilitates collaboration between machine learning experts, designers and end-users. Source: [Françoise et al., 2021]

Marcelle relies on an architectural model to design human interactions with ML. The architecture model is built upon a modular collection of interactive machine learning **components** with a unified interface, that can be **composed** to form custom processing pipelines and user interfaces. This component-based architecture is **extensible** and facilitates **reuse** of interaction techniques across projects. The architecture is built over **web technologies** to facilitate collaboration, and supports **sharing** of applications, data and models.

Design principles

Component-based architecture

Building IML applications require assembling interactions to manipulate machine learning concepts and objects. These objects of interest are highly heterogeneous and relate to various activities, such as the ones reported by Dudley and Kristensson [Dudley and Kristensson, 2018]. Depending on the activity, users might need to operate upon various objects: data, algorithms, parameters, models, predictions, explanations, etc. In addition, interactions can be dramatically different depending on users' expertise and intents. Thus, their actions upon the ML pipeline need to be supported by a custom arrangement of interfaces.

Components are the building blocks of Marcelle. They embed the state, logic, and interaction for particular tasks and possess a minimal interface enabling visualization and communication with other components. A component is essentially a JavaScript object that (1) exposes a set of reactive streams that can be processed by other components and (2) provides methods to display the component's graphical user interface in the DOM. Components are versatile in scope and can address various tasks, including data acquisition, data management and storage, models, visualization tools, and standard GUI widgets.

Components often provide a graphical user interface, or view, that can be displayed on demand in a web application. Examples of views of components are included in the dashboards presented in Figure 2.12. Views only communicate with the component using streams: they are reactive to changes and push events into the component's streams. This mechanism separates the view and the component's processing. In other words, a component remains functional in a given pipeline even if its view is not displayed.

Interaction-Driven Pipelines

IML applications involve custom workflows where user interactions trigger various types of processing. Therefore, it is essential to let developers create custom pipelines specifying complex relationships between the user's actions (e.g. capturing a new instance) and the resulting processing (e.g. adding it to a dataset, training a model, updating predictions, etc.). Reactivity is key to handling diverse workflows where heterogeneous event streams must be interconnected.

Marcelle use reactive pipelines, which give developers explicit control over the information flow. Reactive pipelines in Marcelle relies

on reactive programming [Bainomugisha et al., 2013], an event-driven paradigm that is well-suited for the development of IML applications. It facilitates the creation, filtering, transformation, and consumption of asynchronous data streams that propagate changes over the pipeline. Marcelle's implementation of reactive streams relies on Most.js [mos].

Composable Interfaces

Workflows encompass two main facets: the specification of reactive pipelines describing the relationships between various objects and actions, as described in the previous section, and the visual arrangement of components in the end-user interface. In their review of user interface design for IML, Dudley and Kristensson [Dudley and Kristensson, 2018] underline that while there exist common elements, the design of IML interfaces varies considerably according to the data and application.

Since components provide their views, creating user interfaces tailored for a particular application or user is straightforward. Developers can mount any component to a given element in the DOM. To simplify interface design, Marcelle provides two high-level mechanisms for building user interfaces: Dashboards and Wizards. Dashboards provide applications with multiple pages displaying collections of components. The resulting interface is similar to Tensorboard [Wongsuphasawat et al., 2018]. Wizards are dedicated to the creation of walk-through guides for beginners. Wizards are inspired by Teachable Machine's training wizard that walks users through the training of a machine learning model [Carney et al., 2020a]. Marcelle wizards are flexible and allow developers to specify what components should be displayed at every step.

Data Persistence and Communication

In a collaborative scenario involving users with diverse levels of expertise in ML, it is essential that the objects of various types contained in the application are shared among collaborators: datasets, annotations, models, predictions, or logs.

While reactive programming facilitates real-time data communication, most scenarios require data persistence to store parts of the application's state. For instance, the training data provided by the user should persist, even when changes to the pipeline are made. Marcelle can instantiate flexible data stores with various backends: data can be stored in the browser's local storage or on a remote server. Choosing the backend location only requires passing a URL to the data store. Developers can create different backends to customize where different

objects are stored. Data collections can be created on the fly to store custom information when relevant, some of the states of the application (for instance, the model's parameters), or session logs of the user's interactions.

Interoperability with ML Libraries

Machine learning practice now relies on a key set of programming languages and libraries widely used among researchers and engineers. Among them, Python is particularly popular, with libraries such as Scikit-Learn [Pedregosa et al., 2011], Tensorflow [Abadi et al., 2016] or Pytorch [Paszke et al., 2019], to name a few. The architecture needs to provide interoperability with machine learning frameworks in order to be used by machine learning experts. It is essential to provide an interface to communicate data and models between Python programs and components. For instance, enabling access to data stores from third-party programs would help create bridges between programming environments.

Training and running inference on ML models in web browsers is possible with dedicated JavaScript libraries. Using Tensorflow.js [Smilkov et al., 2019], it is possible to make real-time predictions with potentially large models with several million parameters. Marcelle's dataset architecture is optimized for training, using asynchronous iterators that can stream and process data lazily. Yet, the limited computing power of current web browsers harms scaling to larger datasets and models. Marcelle partially supports interoperability with standard machine learning frameworks in Python. Interoperability gives Marcelle the capacity to scale easily according to the developer's computing resources. The Marcelle Python package can be used to interact with a backend server, with reading and writing access to data stores.

To facilitate extensibility and reuse, Marcelle comes with a Command-Line Interface (CLI) to generate new projects, custom components, and backends. The documentation, source code, API, and examples are available on https://marcelle.dev/.

Marcelle's development context

Marcelle was developed at the same time as the course of this thesis. It is not part of the thesis contributions since it was initiated and implemented by Jules Françoise and Baptiste Caramiaux. However, the experimental requirements of this research steered its development and contributed to shaping the concepts around the toolkit described in the corresponding publication [Françoise et al., 2021]. In return, Mar-

CELLE greatly supported this research. The two applications used in the user study of this thesis were developed with MARCELLE which became a major tool for conducting the human-centered research of this thesis. Its composability enabled it to iterate the design of experimental applications just a few weeks apart.

MARCELLE was initially created to support the development of master student's projects for a class on Interactive Machine Learning. With the covid-19 pandemic crisis, the shareable characteristics of MARCELLE have made it a tool of choice for developing remote workshops and user studies involving novices in a machie teaching task. These requirements steered the development of a database backend that could log participants' interactions on a server. The application of the second user experiment presented in section 4.4 also logs participants' answers to tests and questionnaires. It also steered the development of several components, such as the drawing canvas enabling people to create their own training data and accessible uncertainty visualization described in section 3.3.

MARCELLE AND INTERACTIVE MACHINE TEACHING

MARCELLE does not yet implement all design concepts suggested by the recent literature on IMT. For instance, it does enable users to compose and decompose semantic features natively, nor create schemas i.e. chaining models inputs and outputs in a graph.

However, its reactive programming scheme could allow the library to extend toward these characteristics since output streams from a model could be easily composed with other input streams of another model. Furthermore, reactive programming foster short iteration cycles between human teachers and machine learner, and explanations for image classification starts to be incorporated into the toolkit[5].

[5] https://demos.marcelle.dev/gradcam-transfer/

The main benefits of MARCELLE for IMT lies in the back-end implementation to share data and models across different collaborators. For example, PICL [Ramos et al., 2020] includes many desired IMT properties and aims to be general across subject-matter experts but is still limited to the processing of a large collection of unlabeled text documents, which I assume could be beneficial to lawyers more than medical doctors that might deal with image documents, for instance. MARCELLE is, for the moment, limited to images and raw data. However, MARCELLE could support IMT design by allowing rapid design iteration on interfaces suited to various collaborators' expertise. On the one hand, a medical doctor could have a personalized collection of com-

ponents to evaluate the model outcomes, correct models explanations, and add new data to the training set. On the other hand, an ML expert could have dedicated components for performance analysis and the choice of model parameters. However, both the medical doctor and the ML practitioner could work on the same objects i.e. datasets, explanations, and models. Marcelle thus offers interesting perspectives to the IMT field through the rapid design of interfaces suited to the expertise of different collaborators, which might shape the way people are willing to teach the system.

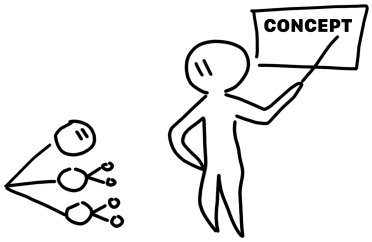
Finally, Marcelle is not specifically designed for running DNNs. However, several of Marcelles' contributions facilitate the use of DNNs in IMT systems. First, *Marcelle* proposes a MobileNetV1 component used for applying transfer learning in the browser. Multi-layers Perceptron can be appended to the MobileNetV1 features to speed up training and predictions on images. Second, the interoperability with python enables the training of more computationally heavy models using local infrastructure.

2.4 Summary

This chapter gives an overview of Interactive Machine Learning (IML) research and its application to specific roles and expertise (ML experts, designers, developers, artists, scientists, and the general public). It then defines and illustrates the specificity of Interactive Machine Teaching (IMT) regarding its goals, the specific role of end-users (teachers), and the desired interaction between human teachers and machine learners. Finally, it presents Marcelle, a toolkit for composing IML workflows and interfaces, and how it both benefited and contributed to the conduct of this research. Finally, we discuss how Marcelle could scale to IMT design concepts and collaborative model-building processes.

Chapter 3

How do people teach a machine?



This chapter investigates how people teach machines. It focuses on the ways novices handle learning algorithms, what they understand from their behavior and what strategy they may use to "make it work". The first section presents the science popularization context in which the research is anchored. The following sections then present an experimental study in which participants performed individual and realistic ML-teaching tasks using a think-aloud protocol. The study investigates participants' strategies and (mis)understandings through incremental interaction with a sketch recognition application called Marcelle-Sketch*, with which participants can incrementally curate and label drawings to train the classifier.*

***Contributions**: I designed and conducted the study presented in section 3.4 under the advice of Baptiste Caramiaux, Wendy E. Mackay, and Frédéric Bevilacqua. I led the analysis of the results with the help of Baptiste Caramiaux, Jules Françoise, and Frédéric Bevilacqua for the cross-validation of the thematic analysis. Jules Françoise developed an early version of Marcelle a few months before* Marcelle-Sketch*. He built* Marcelle-Sketch *and set up the data collection and extraction database for the Twitch workshop and remote individual study.*

We know little about how novice users understand learning algorithms: how do they interpret the system's behavior and understand which strategies they would use to convey concepts to an ML model?

Exploring how general public interacts with learning algorithms is important: First, it can offer us insights on new guidelines for designing rich interactions with ML based systems, following an important line of previous research in the field [Stumpf et al., 2009a, Yang et al., 2018b]. Second, it can bring the technology closer to people such as empowering them in their activities, as described in both IML and IMT fields [Amershi et al., Simard et al., 2017, Lee et al., 2019], and fostering ML democratisation. Third, it can foster ML education [Fiebrink,

2019]. Finally, gaining insight into how the general public interacts with machine learning systems, and in particular the *learning* part of the process, has the potential to increase our understanding of "*machine behaviour*" [Rahwan et al., 2019], and highlight the contextual and socio-cultural influences of Human-ML (and Human-AI) interaction.

In order to explore how novices can train an ML system, we focus on the specific use case of a sketch-based recognition algorithm. In this machine teaching scenario, the goal is to train a recognition system by drawing sketches associated to a set of categories. The system is incrementally trained and the predictions produced from drawings are used as inputs to monitor its accuracy. This chapter is interested in (1) **identifying novice teaching strategies** for an image recognition algorithm; (2) **investigating novice understanding** of the machine behavior;and (3) **highlighting the socio-technical implications of engaging end users with ML.**

The core contribution stems from an experimental study inspecting the use of Marcelle-Sketch by novices. We present a set of quantitative and qualitative findings about users' teaching strategies and users' understanding (or misunderstanding) of the system's behavior, discussed in chapter 6.

3.1 Context and design motivation

This work is anchored in a science popularization context that originated from a collaboration with the association *Traces*, a think-and-do, nonprofit group interested in science, its communication, and its relationship with society[1]. I participated in three different science popularization events initiated by the Traces association.

[1] https://www.groupe-traces.fr/en/traces/

In October 2019, the association *Traces* held a workshop on "Educating Artificial Intelligence" at the TURFU festival in Caen[2], in which we were invited as "AI experts". We presented our research and conducted a short workshop in which participants taught a sketch classifier with categories of their choice. Participants had diverse profiles, from high-school students to jobseekers. The application was a prototype I developed using a Max/MSP front-end communicating with a python server.

[2] https://turfu-festival.fr/

In March 2020, soon after the first covid-19 lockdown, *Traces* organized weekly virtual sessions addressing a wide range of scientific topics to

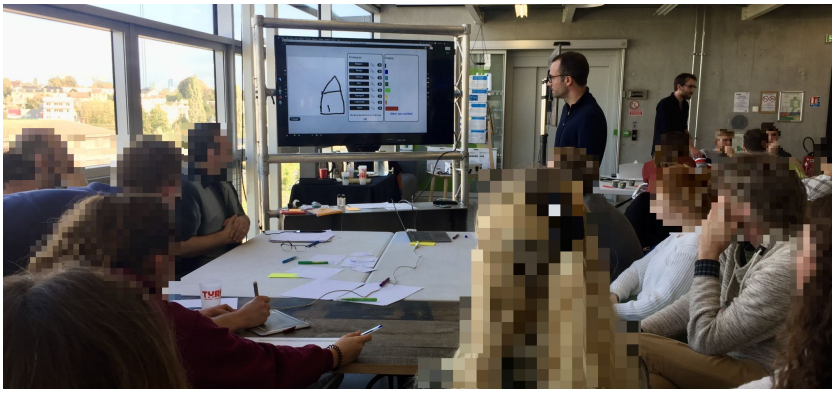


Figure 3.1. Workshop in the TURFU festival in Caen, using a sketch-based IML prototype.

the general public. We collaborated with them for the first session of the series on Artificial Neural Networks. The session was held online on the Twitch streaming platform. We led a remote workshop

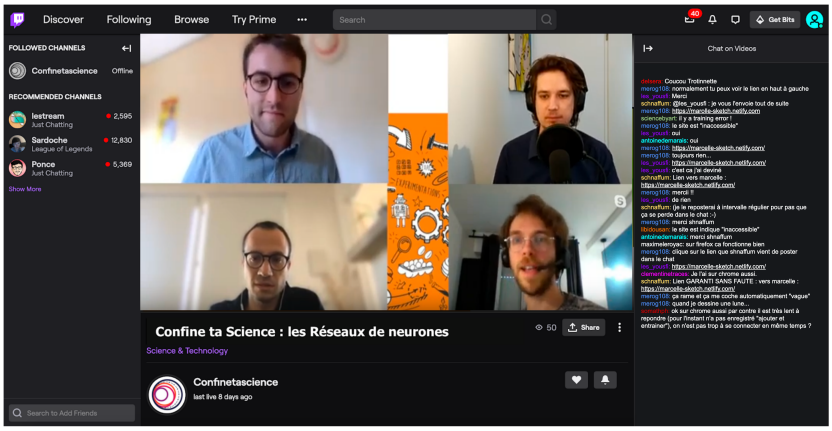


Figure 3.2. Virtual workshop held on the Twitch platform during the covid crisis.

in which participants used Marcelle-Sketch, a shareable IML tool in which they can also teach the system to recognize sketches they drew. We specifically designed and implemented Marcelle-Sketch for this session, which is further described in the next Section 3.3. The application runs in a web browser and is available online [3]. This second workshop was a pilot for designing our individual user study presented in section 3.4.

[3] The first version used in the workshop: https://marcelle-sketch.netlify.app/

In May 2021, the Traces association invited me to two workshops led in a high school in Clichy, in the suburbs of Paris. With a member of the Traces association, we led a workshop discussing AI and Machine Learning. Students used a Marcelle application to train a classifier using images from the computers' webcam. Samples are easier to create using a webcam rather than sketches. This, it appeared to be

better suited for the workshops that had significant time constraints. However, webcam-based IML systems are harder to share since not all devices are equipped with webcams, and Marcelle is not suited for mobile phones yet.

Figure 3.3. Lycée Newton in Clichy, in which the third round of workshops was held in may 2021.

Besides understanding IML systems in a pedagogical context, these workshops were an opportunity to probe peoples' literacy in ML (and, more broadly, in AI). The general public might struggle to envision the scope of action of these technologies and the concrete impact on their lives. For example, high-school students in the third workshop had to define AI before the workshop. Their definitions are listed aside, and we can extract two emergent characteristics emerging from these definitions.

Participants seem divided on whether AI is an autonomous entity or relies on human labor and data. I believe that IMT activities involving participants in data collection and labeling develop peoples' understanding of the ML dependence on labeled data and human labor.

Participants sense the social implications of these technologies, but their scope of action remains unclear. This is particularly striking in definition 3, which cites two applications without clearly acknowledging how AI is involved. Again, involving participants in the training of an ML model could help demystify the application scope of ML by

Definition 1: "AI refers to many developments related to new technologies. They are recent and have experienced a boom from the 21st century. These new technologies are mainly associated with intelligent cars, machine learning, robots. The objects designed are generally autonomous and independent."

Definition 2: "Autonomous algorithmic program created by humans capable of learning and adapting to time!"

Definition 3: "It is an autonomous and discrete technology, allowing to replace some human activities. It brings advantages like Elon Musk with his electric car Tesla which allows improving road safety (and to pollute less) but also disadvantages like in agriculture which causes the loss of jobs."

Definition 4: "Program created by humans allowing to recreate a computer brain. It can execute complex tasks faster than humans. It may have the ability to learn over time."

Definition 5: "Computer science based on several algorithms aiming to perform tasks more efficiently and complex than humans. It is based on autonomous learning and works with data."

Definition 6: "Artificial intelligence is knowledge developed on a specific domain, virtual or material, which can potentially be improved. AI is used for the benefit of humans."

Figure 3.4: Definitions of AI give by high school students

anchoring students' knowledge into a concrete experience: the training of a classifier on real-world data. I believe it is an engaging activity that can be a gateway to the AI and ML disciplines.

Overall, these workshops demonstrate the engaging and playful potential of IMT for the general public, but longer workshops would be necessary to engage participants in real reflection on how they imagine using and designing tools that they could teach. For researchers, shareable and easy-to-use IML systems offer opportunities to probe how interactive machine teaching can shape peoples' literacy about ML and AI. For these reasons, the applications used in the three workshops were designed following three important requirements:

1. People should be able to produce their own data to teach the system;

2. People should receive immediate feedback about the model's predictions and uncertainty;

3. People should be able to use the application anywhere and easily.

With the first requirement, we aim to involve users in generating and curating the training examples. We are interested in studying the teaching strategies that emerge when users are free to change the input data in response to the system's outcomes. Except for the third workshop, we use drawn sketches as inputs because they do not require specific expertise or hardware, and they are personal.

Second, people need to be able to interpret the model's predictions. Model predictions always embed uncertainty which is also important to convey to the users. A common feedback strategy displays likelihoods, i.e. values between 0 and 1, conveying the confidence level that the input instance belongs to each class. In addition to likelihoods, we use another approach that estimates this uncertainty using model ensembles. A further explanation of model uncertainty in deep neural networks is presented in Section 4.2.

Third, our goal was to inspect the real-world use of the system by novice users. As such, we brought particular attention to designing an application that can run online, which is easy to use. This third requirement was crucial for the remote workshop held on Twitch and drove the development of the Marcelle toolkit for producing shareable web-based IML applications equipped with remote backends for accessing interaction logs.

Altogether, Marcelle-Sketch is thought of as a tool to probe novices' teaching strategies and understanding of a sketch recognition system and will be used for our remote and individual user study presented in section 3.4.

3.2 Exploring machine teaching with the general public through the remote workshop led on Twitch

In this section, we present in more detail the workshop we conducted during the live session on the Twitch platform in collaboration with the *Traces* association. The analysis of participants' interactions informed the design of the individual remote study presented in section 3.4 and the design iteration conducted on Marcelle-Sketch presented in 3.3.

The live stream lasted about 90 minutes and was divided into three parts. The association moderators started with an introduction to artificial neural networks that lasted around 20 minutes. Then, we conducted the machine teaching workshop for 40 minutes using the Marcelle-Sketch application presented in Section 3.3. Finally, we answered questions from the audience asked via the chat for 20 minutes.

Procedure and participants

After introducing neural networks, we started the workshop by presenting the interface. We sent them a link to the application in the chat, and participants opened it in a new tab. If they had questions, they could communicate with them on the chat. One moderator gave a live demonstration of the application while a researcher was giving the explanations.

We chose a pre-defined set of categories to structure and focus the observations on the teaching strategies. Participants could not train on new custom categories. Enabling the creation of new categories would have made difficult the comparison between participants' strategies, both in the pilot workshop and the study (presented in the following section). The number of categories was fixed, and their labels predefined: "Moon", "Hat", "Wave","Cheese" and "Time". The model was initialized with random parameters (except for the MobileNet embedding) at loading, and the training set was empty. We asked the participants to train the system until the model was accurate and confi-

dent about the predictions for each category. We gave them 20 minutes to perform the task at their own pace. After that, we explained how to share their project publicly. Then, we asked them to load a model from another participant and explore it with their own drawings. Finally, we invited the audience to answer an online questionnaire about the training process, posted in the text chat.

At the beginning of the workshop, 160 people were connected to the stream live. The number decreased during the session until reaching 84 people at the end of the session. Among these participants, 22 participants made their Marcelle-Sketch project public, i.e. we could analyze the data of 22 participants. 7 participants answered the online questionnaire.

Data collection and analysis

We analyzed participants' use of the system afterward, by collecting images after every stroke they did, including all the predictions from the classifier and the label they chose for the training data. We removed three projects from the whole set of projects that were submitted twice and kept only the projects with at least one drawing per category. Eventually, we kept 14 projects from 14 participants over the 22 projected submitted. The data included the images (png format), the timestamps, the predicted/trained category, and the MobileNet network features, all stored after each stroke made on the interface. Our analysis focused on the order in which categories are trained, the proportion of discarded images, and the images' variability.

Insights and limitations

The analysis highlighted that most participants iterated quickly across categories when training. On average, they did less than two consecutive drawings of the same category before moving to another category. We observed that most participants included all their drawings in the training set. Few participants discarded some of their drawings. Most of the discarded drawings were examples of existing categories that might have helped participants assess if the model had effectively learned previous representations. The remaining discarded drawings were off-category drawings that may have been occasionally used out of curiosity as a way to challenge the algorithm without specific expectations on its outcome. Finally, we found that participants used different variations of the drawings for each category, including variations in representations of the concept (for instance, clocks and hourglasses to represent the "time" category) or transformations such

as orientations, colors, and shapes.

The workshop allowed us to collect rich data in a non-controlled experimental context. However, it also brought limited insights regarding our second research question that focuses on how the participants understand the system during a realistic teaching task. We decided to conduct an experimental study, using a think-aloud protocol in individual sessions, to investigate further the underlying choices and decisions behind the observed behavior and how participants became aware of ML-based systems.

3.3 Marcelle-Sketch: Application overview and design iteration

Marcelle-Sketch was developed with an early version of the Marcelle toolkit presented in section 2.3. Significant changes have been made to the toolkit since then. Furthermore, we iterated on the design of Marcelle-Sketch after the remote workshop held on Twitch. The second version was used in the individual remote study presented in section 3.4. The application is available online [4] and illustrated on figure 3.5.

[4] https://marcelle-sketch-v2.netlify.app/

Application overview

Marcelle-Sketch is a dashboard composed of two panels, as depicted in Figure 3.5. The left-side panel is dedicated to inputs. It exposes a white canvas where users can create drawings. It also allows for data management such as dataset download or upload. The right-side panel is dedicated to prediction, training, and data visualization.

The workflow is as follows. The user starts drawing a line ("sketch input") and releases the mouse button. Predictions are automatically updated (chart bars) and the prediction uncertainty (gauge). The user also receives feedback on the predicted label (drop-down menu below the gauge). If the user wants to correct the prediction, they can click on the drop-down menu, select the correct label, and then click on the button to update the training set and launch training. Training is fast (a few seconds). Once the training is done, both the prediction and uncertainty are automatically updated using the newly trained model. The user could also choose not to add the drawing to the training set and keep adding elements to their drawings, inspecting the changes

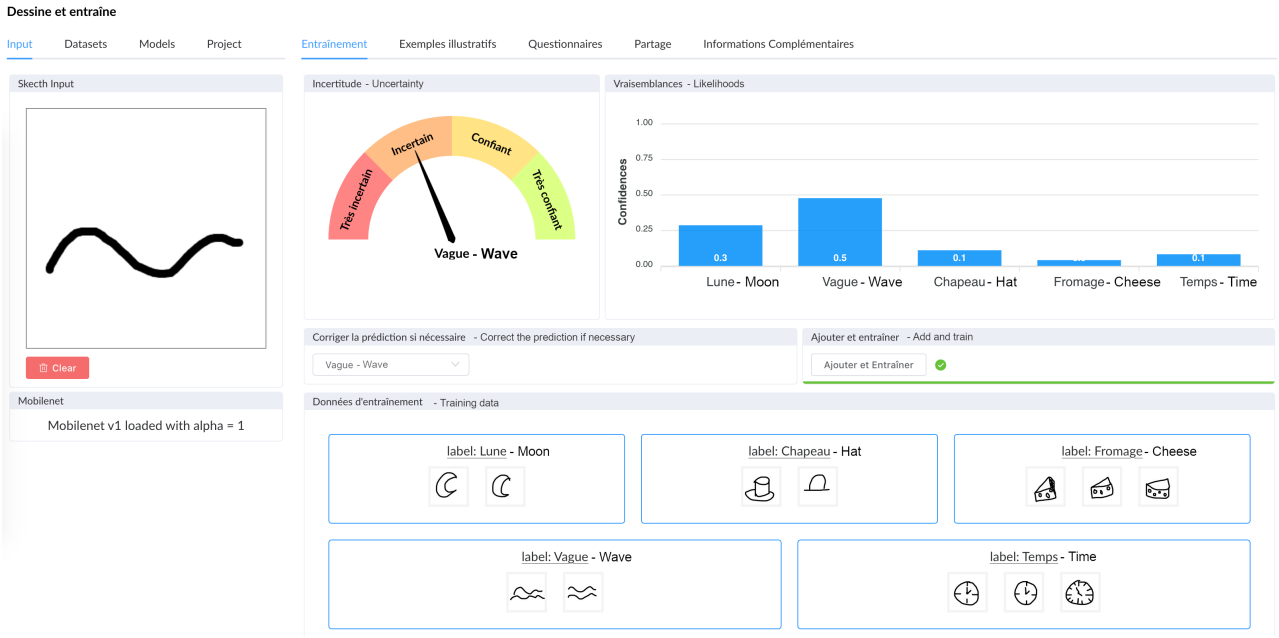


Figure 3.5. Application interface used in the think-aloud study. Miniatures of the drawings are displayed on the main screen (bottom right component). We removed the history component and the possibility of changing the drawing pen's line width and color for more controlled data.

in predictions and uncertainty.

The application is built using *Vue.js*, a JavaScript frontend framework. Each component displayed on the interface is a Vue component and is reactive. If a change is made on data (e.g. a new sketch or a new prediction made by the model), the components' display is updated automatically. The server was built with *Node.js*, and the data were saved in a MongoDB database.

Machine learning pipeline and technical features

Marcelle-Sketch is designed to allow for online and fast learning. The key technical features of the applications are presented in this section. The machine learning pipeline is divided between an encoder used to extract features from the raw image representing the user's sketch and a classifier (built on top of the feature encoder), which the participant trains.

Machine learning pipeline and transfer learning

The application uses MobileNetV1 as a pre-trained deep neural network [Howard et al., 2017], which provides embedding suited for image classification. Its architecture uses depth-wise separable convolutions to build lightweight embedding. The weights are initialized randomly when the application is loaded, the training is then incremen-

tal, and only the learning rate is reset at each update. As explained in appendix A, the use of transfer learning through MobileNet allows a simpler classifier to be trained quicker (a few seconds), using minimal data (below 100 instances), which is critical in our scenario. Then, the method can be made more robust by assessing the model uncertainty, which is the second technical feature presented in the next section.

Classification with uncertainty estimation

Technically, one originality of our approach is the use of an ensemble of models, called *Deep Ensembles* [Lakshminarayanan et al., 2017] to improve model performance under limited data and to allow uncertainty estimation. The *Deep Ensembles* approach was presented in section 4.2 and involved to train a set of N distinct classifiers (initialized randomly). Each classifier is trained independently, and their predictions are combined to produce the final prediction. In the proposed system, we built an ensemble model comprised of 5 Multilayer Perceptrons (MLPs), initialized with random weights on the top of the MobileNet encoder. The ensemble is trained in parallel with the user data: the 5 MLPs simultaneously learn the mapping between the MobileNet features and the 5 pre-defined classes.

The benefit of having an ensemble of models trained in parallel is the possibility to compute an estimation of the model uncertainty over the predictions. We used variation ratio as an estimator of the prediction uncertainty [Beluch et al., 2018], defined as the number of models that agree on the same class divided by the number of models. In other words, the variation ratio can take five values of uncertainty: between 1/5 = 0.2 (all the ensemble models disagree on the prediction) and 5/5 = 1 (when all the models in the ensemble agree on the prediction). We mapped its values to four categories: "Very uncertain" ($ratio \leq 0.4$), "Uncertain" ($ratio = 0.6$), "Rather confident" ($ratio = 0.8$) and "Confident" ($ratio = 1$). This uncertainty is displayed to the user on the gauge of the component "*incertitude*" on figure 3.5.

3.4 User study: think-aloud individual teaching sessions

We conducted a remote think-aloud protocol with novices (in ML and CS) with the following objectives: **(1) identify novices' teaching strategies** of a sketch-based recognition algorithm; and **(2) investigate their understanding** of the machine behavior. The teaching task is similar to the pilot workshop and consists in teaching a classification algo-

rithm from scratch to recognize hand-made drawings in Marcelle-Sketch. The methodology employed in this study borrows from the structured observation approach [Garcia et al., 2014, Mackay, 2014] introduced in section 1.3. Indeed, we do not attempt to test an hypothesis but gather observational data to increase our understanding of novices' behavior and understanding when placed in the situation of machine teachers.

Participants

We recruited 12 participants with limited to no knowledge in machine learning or computer science. We recruited participants by email among contacts of the association and from the university's students, avoiding scientific or technological profiles. Among the 12 participants, 7 are female, and 5 are male. 7 participants are aged between 18 and 29, 1 between 30 and 39, 3 between 40 and 49 and 1 between 50 and 59. Participants graded their prior knowledge about image recognition systems in the pre-questionnaire about their knowledge. 6 participants answered that they are novices, 4 participants are "little informed", 1 participant is "informed,' and 1 is knowledgeable.

Setup

We used an open-source video conferencing platform hosted on a secure server to communicate with the participants. The video conferencing platform can be accessed from the browser. We asked the participants to share their screens at the beginning of the session. We video-recorded their shared screen while they were training the model. We used the computer microphone to record the audio from the video-conference application. The participants performed the task on their own computer, using Marcelle-Sketch in their browser. The application was linked to a server and a database to collect data, such as participants' drawings and models. From the version of Marcelle-Sketch used in the pilot, we removed the possibility to change the color and the width of the pen. Participants did not often use it, and it allowed us to reduce the variability and better compare the teaching strategies. The questionnaires were created with an open-source platform called Framaforms and shared with the participants through a link.

Procedure

When participants log in to the video conferencing platform in the browser, the experimenter starts by explaining the general structure of

the experiment. During the session, the participants are told that they will have 30 minutes to train an image recognition algorithm to recognize drawings that they will create, each drawing belonging to one of the predefined categories. Then, a link to the Marcelle-Sketch application is sent to the participants. In the application, the third tab of the interface is a page where appears the link to the pre and post-questionnaires. Participants are asked to fill out the pre-questionnaire. The purpose of the first questionnaire is twofold. First, we want to inspect participants' knowledge about image recognition algorithms. Second, it serves as a primer to encourage them to think about how image recognition algorithms work. Participants are asked to share their screens once the pre-questionnaire has been filled out. The main teaching session comprises three steps:

1. *Explanation of the task and interface*. The task is explained to the participants to teach the algorithm to correctly classify drawings that they make with the mouse into pre-defined categories. We use the same categories as in the workshop: "Moon - Lune", "Hat - Chapeau", "Wave - Vague", "Cheese - Fromage" and "Time.- Temps". Then, we explain each interface component to the participants, and we start recording the session.

2. *Think-aloud teaching phase*. Participants have 30 minutes to teach the model. During this training phase, we ask them to think aloud. If the participant stops talking for a few minutes, the experiment conductor reminds them to comment on their thoughts.

3. *Think-aloud retrospection on the data*. After the teaching phase, there is a 10-minute phase to encourage the participants to reflect and debrief on the algorithm recognition abilities. Participants are asked to describe: (1) which drawings are correctly recognized by the algorithm and (2) which drawings the model is uncertain about. Like the teaching session, participants are asked to comment on their choices out loud. The screen and audio recordings are stopped after this step.

The study ends with a post-questionnaire, which aims to evaluate how participants perceived the system and how participants' prior ideas about the behavior of an image recognition algorithm evolved after the interaction.

Data collection

We recorded the think-aloud sessions through screen recording. In addition, we collected the datasets made of the intermediate drawings (i.e. drawings after each stroke) and the training set (i.e. drawings used to train the system). We also collected the two datasets built after the training, containing "recognized drawings" and "drawings the model is uncertain about". For those four datasets, we stored drawing as *png* images together with their creation timestamps, the predicted category when drawn (or assigned category when trained), the computed uncertainty, and the features from the MobileNet network. Finally, we collected the answers to the questionnaires stored on the Framaforms platform.

Data Analysis

Quantitative analysis of the teaching process

We computed three measures related to the drawings performed by the participants to teach the model. Our first research question on characterizing novices' teaching strategies motivated these measures. The measures are:

- *The amount of drawings trained* i.e. how many drawings were used to train the system. It relates to the speed at which the participant draws and how often participants want to use a finished drawing to train the model.

- *The variability in the drawings*. We computed a measure of variability within a category using Euclidean distance between pairs of drawings in the feature space, i.e. the output vectors of MobileNet associated with each drawing. We averaged distances between all pairwise combinations of instances within a category (to avoid comparing images from different categories). We then averaged the variability across categories for each participant. Formally:

$$V_{participant} = \frac{1}{5} \sum_{c \in categories} \frac{1}{C^2_{size(c)}} \sum_{X_i, X_j \in c} d(M(X_i), M(X_j)) \tag{3.1}$$

  with $C^2_{size(c)}$ the number of combinations of 2 instances in the category c, d the Euclidean distance, and $M(X)$ the feature vector after passing the input image X into the MobileNet network. To help the reader appreciate the variability across participants, Figure 3.7 depicts the training set of the most variable and least variable participants.

- *The average number of consecutive inputs with the same category*. This measure highlights the sequencing i.e. the order in which participants trained the proposed categories. We display participants' sequencing on the upper timelines on figure 3.10 at the end of the result section.

Quantitative analysis of the model performance

We computed the following measures related to the performance of the trained classifier:

- *The generalization performance* measures how each trained model can generalize beyond a participant. We used the final trained model of each participant. We then computed an accuracy score on the test set composed of all the training sets from the 12 participants.

- *The personalization performance* measures how well the model can fit a participant's data provided during the training session. We also used the final trained model of each participant. We then computed an accuracy score on a test set composed of all finished drawings of the participant (that are used to train the model or not). We annotated the finished images (images before the participant "clear") by hand, discarding errors or involuntary strokes.

The performance scores are used as indicators of the model abilities rather than a quantification of the task completion. Participants were not asked to improve the generalization of their model when we introduced the task to them.

Qualitative analysis of the verbalizations

To analyze the verbal elicitation from the participants, we applied thematic analysis [Braun and Clarke, 2006] to code and categorized the transcribed audio recordings. Two authors first labeled each meaningful verbalization, describing the participant's actions or thoughts. From these labels, we created a set of themes that convey the participant's intent and address our research goals: (1) identify novice teaching strategies for an image recognition algorithm and (2) investigate novice understanding of the machine behavior. The theme created during this phase are:

- 5 themes about the participants' **learning behavior understanding**: *"interpretations and beliefs about the learning behavior of the* system", *"asking oneself about the learning behavior of the system"*, *"misunderstanding"*, *"the participant felt the system could learn a drawing successfully"*, *"the participant felt the system could not learn a drawing success-*

fully";

- 2 themes about the participants' **teaching decision**: *"justification of an action according to previous ones"*, *"organisation and structure of the overall session"*;

- 2 themes about the participants' **teaching intentions** *"evaluation of previous images learned"* and *"exploration of new drawings"*.

In addition to the thematic analysis, we aligned the verbalization to the drawings mentioned within them to understand the course of events and context better. Then, the authors coded the verbalizations again according to these themes. The first author of the paper coded all participants' transcriptions, and three co-investigators coded four participants each. We gathered the codes and discussed their alignment. We categorized the 710 quotes from the 12 participants over the 9 themes mentioned above.

We finally kept the quotes where a clear agreement could be found between annotators, so approximately 350 quotes. The study, the transcriptions, and the analysis were conducted in French. The translation to English was only made to report the results. Note that the neutral pronoun is identical to the masculine pronoun in French. We then decided to keep the neutral pronoun every time the participant referred to the system.

3.5 Results

In this section, we report the findings resulting from (1) the qualitative and quantitative analysis of participants' teaching strategy and (2) the qualitative analysis of their understanding of the machine's learning behavior.

Analyzing teaching strategies

In this section, we present our findings related to the first research question on identifying teaching strategies by novices and their relationship to model performance. The results in this section are primarily quantitative. They are complemented with quotes from the thematic analysis, which allows us to describe better participants' intentions about their strategy (when verbalized).

Novices adopt contrasting strategies

We analyzed the teaching strategies by looking at three measures informing on the teaching process: the number of drawings trained, the variability infused in the inputs, and the adopted teaching sequencing (see Section **??**). Figure 3.6 depicts each participant within this teaching strategy space.

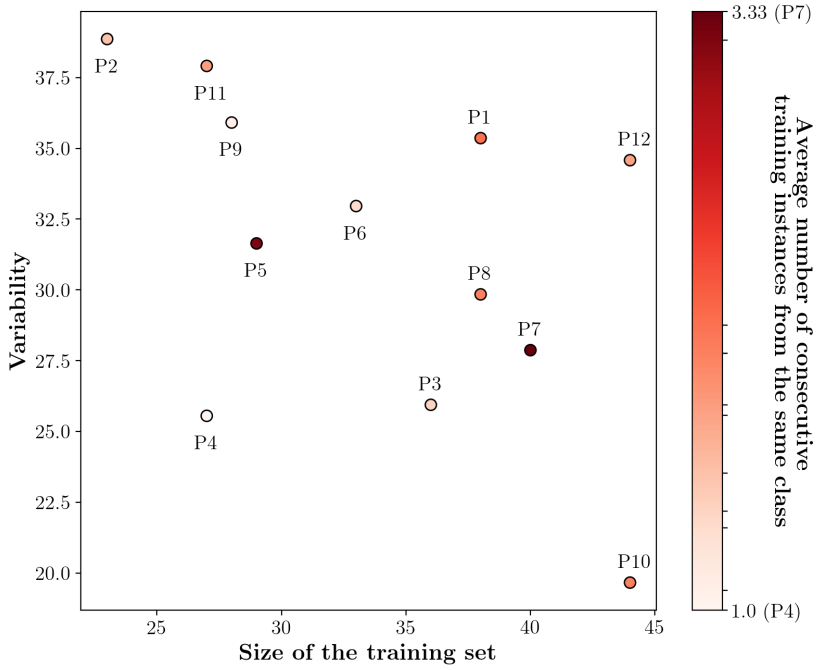
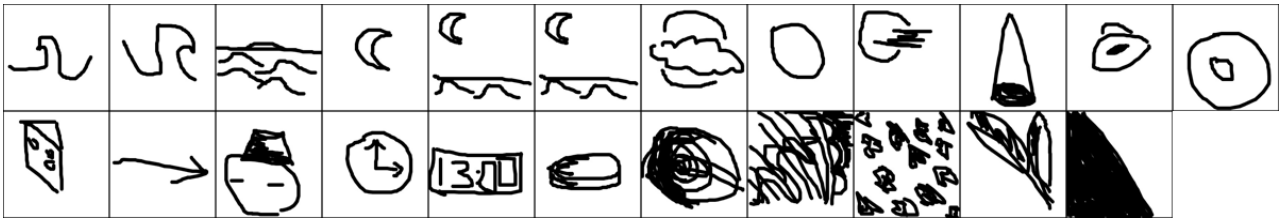


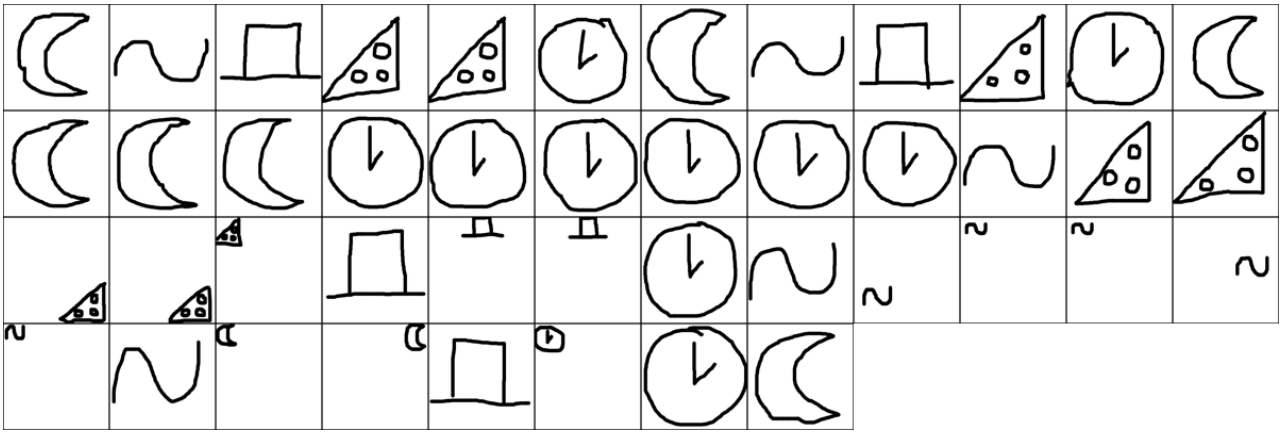
Figure 3.6. Teaching strategy space: Variability (y axis) according to training set size (x axis) and sequencing (color map).

We first investigated whether these dimensions provide insights on complementary aspects of teaching strategies. We computed the correlations between these three dimensions and found that there are not significant, meaning that each measure represents a dimension of the teaching strategies adopted by participants. In addition, we found that participants are well distributed in the space. Within the 2-dimensional space created by the dimensions *variability* and *size of the training set*, the two extreme cases P2 and P10 suggest that low variability is more often related to simple shapes in the training set. The training set of P2 and P10 is depicted on Figure 3.7.

We also found that the participants adopted different teaching strategies by analyzing how they sequenced the training instances. We found that the number of consecutive training drawings from the same category spans from 1 to 3.3 (see Figure 3.6). For instance, P4 never trained the same category with two consecutive drawings (leading to a consecutive rate of 1). By contrast, P5 and P7 have consistently drawn



(a) Training set of participant 2



(b) Training set of participant 10

Figure 3.7. Most variable (a) and least variable (b) training set among participants.

on average more than 3 drawings in a row from the same category. The average number of consecutive drawings from the same category is 1.9. This result highlights the spectrum of strategies from focusing on one category at a time (using several drawings) to constantly changing the training category. Importantly, participants did not explicitly state that they used a sequencing strategy.

> **Finding**: Participants adopted heterogeneous teaching strategies in terms of training size, variability, and sequencing, which underline the lack of means of the classifier on the actions to be taken to train it.

Impact of the variability on system performance

We consider two types of performance indicators: generalization performance and personalization performance (as described in Section 3.4). In this section, our goal is to link participants' strategies, described in the previous section, to these notions of system performance. However, this study is a structured observation conducted out-of-lab settings. Hence, regular frequentist statistical inference methods like ANOVA cannot be conducted and would yield to low statistical power considering the low number of participants.

Data variability tends to be correlated to generalization performance, suggesting that participants that infuse greater diversity in their drawings train a model that tends to better generalize across other participants' data. To a certain extent, this was expected since, in ML, variability is known to be beneficial to generalizability. However, this rule can also be mitigated by the fact that idiosyncratic variability could degrade the performance because fewer correlations within the data can be found.

Interestingly, as a counter-example, P5 created a dataset with low variability and reached a higher generalization score than P9, which created a dataset with high variability.

Figure 3.8 depicts examples from the data provided by these two participants. It shows that P5 favored simplistic, icon-style representations while P9 opted for more complex and idiosyncratic representations. Therefore, *variability*, as considered in this work, does not systematically imply a good generalization score. These results suggest that the nature of this variability is critical.

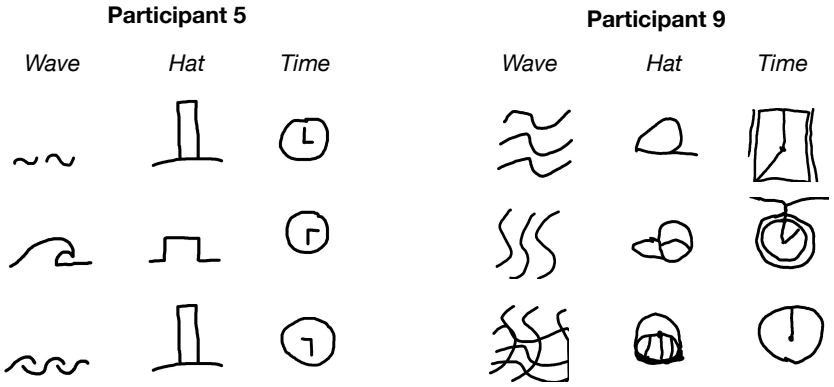


Figure 3.8. Samples of the training set of P5 and P9. P5 adopted a more icon style, whereas P9 opted for more idiosyncratic drawings.

We found that participant 12 is the participant that obtained the best scores in both performance indicators. P12 obtained the best generalization score (accuracy equals 0.40) and the second-best personalization score (accuracy equals 0.82). P12 has the largest training set and one of the highest data variability. P12 managed to create well-separated categories that may be shared across participants. P12 also gradually increased the difficulty of the inputs curated. As a matter of fact,

P12's verbalizations in the theme *"organization and structure of the overall session"* give us information about the dynamic of her teaching strategies. She elicited a precise training policy early in the session to avoid

adding similar instances if they are already confidently recognized. She then updated her decision "threshold" based on the subjective quality of the drawing: «*If it's still confident even if I make an ugly drawing, I want to start training it to be very confident with my ugly drawings. That's going to be my new policy because I see that it can be confident with my ugly drawings*». This process operates as a *curriculum* for the recognizer.

Finding: Variability tends to favor the generalization of the model, while the other dimensions of the teaching strategy do not seem to affect the system's performance. The type of variability, and the fact it might be introduced progressively, plays a role in building an efficient classifier that can handle various representations.

Sequencing affects the model performance and performance perception

The sequencing (i.e. average number of consecutive instances trained with the same category) is not correlated with generalization or personalization performances. However, we found that the first drawings used to train the system are critical to ensuring a good performance. Participants who focused on a single category at the beginning of the session created a model that predominantly predicted this category over the rest of the session. This phenomenon is due to the incremental nature of the training procedure involved in the system. The model is optimizing its parameters according to limited data drawn from a single category. The loss function can then remain locked into a local minimum, blocking the network parameters. The model then requires multiple iterations on new instances from other categories to escape from this local minimum and reach a better optimum.

Figure 3.9 depicts the training sequencing for participants 7, 1, and 8. For each participant, the top line represents the training sequencing (each instance from the beginning to the end of the training and its label), while the bottom line represents the predictions. These participants are the ones who trained at least four images from the same category at the very beginning of the session. We can see that the consecutive predictions remain the same as the first category. P1 succeeded in canceling this effect at about 37% of the session by providing a balanced number of instances to other categories. The effect remains for P8 and progressively disappears between 33 and 60% of the session. The effect seems to persist for P7 until the end of the session. This might be because P7 trained the highest number of consecutive

instances from the same category at the very beginning of the session (9 consecutive "Moon"). Figure 3.10 depicts the same visualization for other participants. We can see that this effect also affected participant 2.

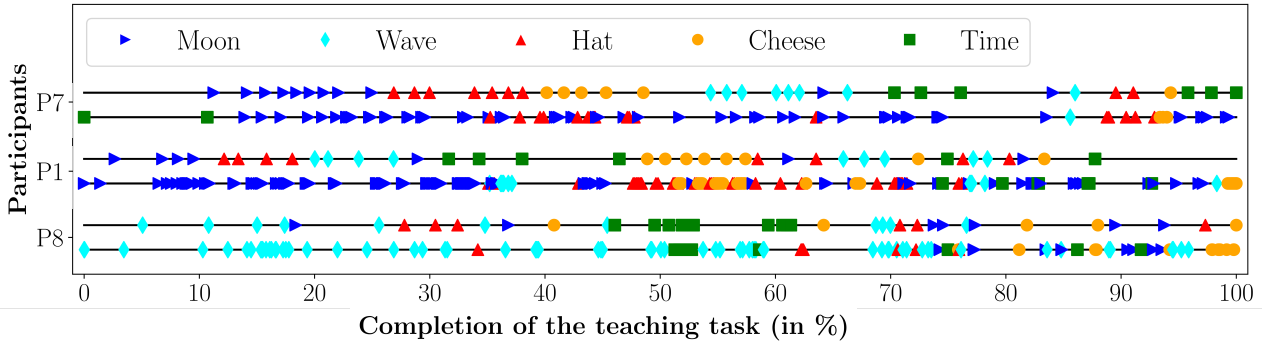


Figure 3.9. Categories trained (upper timeline) and predicted (lower timeline) in chronological order for participants 7, 1, and 8. These participants trained at least four consecutive images with the same class at the beginning, and their predictions were affected for the rest of the session. The x-axis represents the completion of the task (in %).

From the verbalizations related to the themes “misunderstanding” and “the participant felt the system could not learn a drawing successfully”, we notice that P7, P1, and P8 perceived this inertia effect while not necessarily understanding it. Only P1 seems to adopt appropriate actions. Indeed, P7 mentioned two times that “it really likes moon”, while P8 and P1 refer to this effect multiple times: *«But why it still thinks it’s a wave there, I don’t understand.»* (P8), and: *«I change the category because it always refers me to the Moon»* (P1).

Finding: The training sequencing (i.e. the order in which examples are given) has an essential role in incremental teaching, especially at the very beginning of the teaching. The actions necessary to unlock confusing model behaviors are not transparent.s

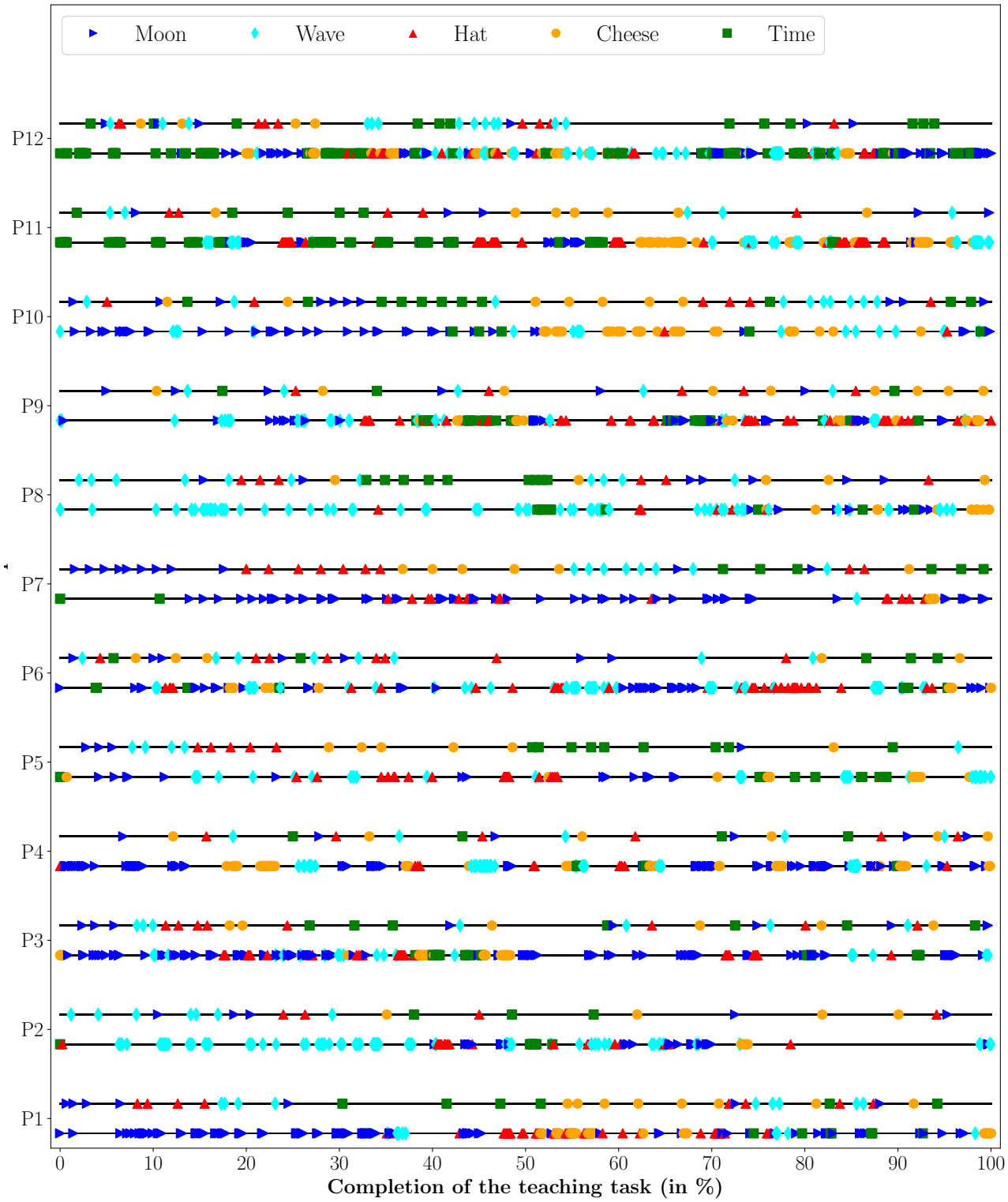


Figure 3.10. Categories trained (upper timeline) and predicted (lower timeline) in chronological order for each participant according to the completion of the task (in %).

Understanding the machine's learning behavior

We now present the results relating to novices' understanding of the system's learning behavior. This section reports qualitative results drawn from the thematic analysis (findings 4 and 6) and the questionnaire (finding 5). The quantitative data (images drawn) were only used during the analysis to give a broader picture of the context.

Participants investigate and teach input feature variations.

We found that some participants became aware of the features that the system takes into account in the recognition process. In a first preliminary analysis, we categorized verbalizations in which participants mentioned variability. They are gathered in the theme "exploration of new drawings" mentioned in Section 3.4. When we categorized the quotes in the theme "exploration of new drawings", we noticed the occurrences of geometric vocabulary (rotations, size changes) and decided to group these explorations as "operations". This group includes the reuse of the same representation for geometrical transformations or duplications. The group "executions" stood out since two different gestures could lead to the same representations. Finally, the "representations" group encompasses all remaining extracted labels. The drawings in this group are all characterized by changes in the composition of the drawing i.e. drawings made with a different organization of the strokes with respect to each other. We built a taxonomy of the different input features that the participants mentioned when introducing variability from this categorization. This taxonomy is summarized in Table 3.1. As we mentioned above, we identified three groups: 1) the *representation* of an image such as the shape, the infilling, the relief (plane or depth), and context (adding contextual details on the image); then 2) the *execution* of the drawing, such as the gesture used to draw; and finally 3) the *operations* on an image such as translation, rotation, duplication (drawing several representations on one image), or change in size. In Table 3.1 we also report participants who used these features and an example from their verbalization. In bold in the table are the participants who intentionally conducted investigations to understand how the system could handle this feature. By contrast, underlined participants are the ones who came up with conclusions from their investigations.

Feature group	Feature	Part. ID	Quote example
Representations	Shape	P6, P7, P9, P10, P11	"Ok so I think it's pretty much all learned now, mostly based on the shapes" (P11)
	Infilling	**P2**, **P6**, P7, **P11**	"I was wondering [...] if it's only the structure that I draw, if it would be detected as a moon even with the color with all the details" (P11)
	Relief	P9	"I think that's a key thing like knowing the difference between 2D ans 3D." (P9)
	Context	**P11**	"Now I'm trying to add more other details rather than just "vague" [...] to see if the machine can still detect the main subject of this painting." (P11)
Execution	Gesture	P6, **P10**	"I thought it was recording the final image, but it's possible that it records every movement I make." (P6)
Operations	Translation	**P8**, **P10**	"Maybe the position didn't change anything. I'm going to put the cheese in a different corner." (P10)
	Rotation	**P7**, **P8**, P12	"First I will try to see if my theory is confirmed, that there is no direction" (P8)
	Duplication	P9	"I tried different methods such as doubling the amount, maybe even tripling, quadrupling, so many many more" (P9)
	Size	**P8**, **P10**	"Does size matter? [...] I do a little clock test depending on the size and it doesn't work at all." (P8)

Table 3.1. Input features that are presumed to be considered in the learning process. Participants that investigated their hypothesis with further inputs are indicated in bold.

From this analysis, we found that participants created new insights on the model mostly when investigating *operations* and *execution*. We assume *representation* features are harder to isolate in order to conduct investigations. For instance, changing the context (adding related representations on the drawing) also affects the general shape of the drawing. Conversely, *execution* and *operations* can easily be isolated and tested on learned representations.

Intentional investigations on the representations were mainly made with "infilling" i.e. participants investigated what happened when they changed inner details, such as texture and color. P6 and P11 both concluded that the color did not affect the prediction only after drawing one or two new colored images that were correctly recognized. They did not perform extensive analysis of this feature and conclude with a partially false claim about the importance of the infilling regarding the shape.

Regarding *execution* and *operations*, 5 out of the 7 participants that conducted investigations generated insights that were in line with the design of the system. P10 did two identical images regarding execution but inverted the direction in which she made the strokes. Based on these tries, she concluded that only the final image is taken into account. P7 investigated rotations and found that the uncertainty decreased when tilting a learned representation: *«When I flipped the hat 90 degrees, it became uncertain. Maybe I didn't notice that for the moon.»*

(P8). Another example is how P10 explored translations and resizing. P10 drew each category with a regular size, and then drew a small "cheese" in the bottom right corner (see figure **??**). P10 trained the model with the transformed representation twice. After placing the cheese in another corner (top left-hand corner), P10 became aware of the translation properties of the machine learner: *«It still thinks it's a cheese [...] when it's not at all in the same corner of the picture, so it must not be the position in the picture»*. Here participant 10 understood that the model is invariant to translations (i.e. position). Then she did the same operation with other categories, but the system kept predicting "cheese". P10 concluded that: *« I first showed the system that cheese could be in different corners, so it understood for the cheese. When I do other things in other corners, it still thinks it's a cheese »*. This case illustrates how participants who actively investigate operations (i.e. transformation of examples on which the model is already trained) may build a more precise mental model about the underlying algorithm and the features it takes into account. In other word, participants explore different model's blind spots [Meek, 2016] and can perceive when an example resolve the model blindness as well as the models' invariant.

Finding: Participants verbalized various features that the system might take into account in the learning, and they tend to discover insights about the system's inner workings when investigating "execution" and "operations".

Participants understood the order in which the examples were given affects the training

In the pre and post-questionnaires, we asked the following question: "According to you, how important do you think the following criteria are for learning the algorithm?". We provided a list of criteria that participants annotated on a 5-point Likert scale (from "not important at all", to "very important"). Using pairwise t-tests, we found that the importance attributed to "the order in which examples are given" significantly increased after the teaching session ($p = 0.011$).

P8 and P12 explicitly expressed doubts about the importance of order during the session: *«Yes, so you'll notice that I didn't take the time to sit down and think [...] without thinking about whether the order in which I draw will have an impact on the algorithm in fine.»* (P8). P5 and P2 became aware of order regarding the wrong predictions following the category they trained: *«If I had started by drawing rectangle-shaped cheeses before the hats, it would have recognized the cheeses well. So it's not that it's badly rec-*

ognized, but it's because I did it in that order!» (P5). *«Oh yes, so everything is a wave. From now on, everything is a wave».* (P2). It is worth mentioning that P2 and P5 have a high number of consecutive examples from the same class, meaning that they mostly focus on training one class after the other. The design choice (incremental teaching) and the phenomenon described in finding 3 (model locked on certain predictions) are probably responsible for the participants' reconsideration of the order effect after the experiment. Participants did not anticipate this effect, suggesting that they were expecting a more intuitive learning behavior from the machine, possibly closer to human learning. P7 said: *«This is the big difference between the machine and humans because we are intuitive. The machine will never have an intuition».* This result shows the need to help novice users consider the order in the interaction by helping them build a meaningful curriculum in the teaching (discussed in the implications for design section 7).

Finding: Participants became aware of the importance of the order in which drawings are provided, which may characterize incremental teaching.

Underlying neural network properties are confusing for novices.

This section studied all the quotes where participants asked themselves questions about the system's behavior or expressed a lack of understanding. The quotes are gathered in the themes *"asking oneself about the learning behavior of the system"* and *"misunderstanding"* introduced in section 3.4. We categorized them according to the source of the confusion. If the majority of the confusions are due to unexpected predictions, 29% of them stemmed from properties of Neural Networks. From these confusions, we built a taxonomy reported in Table 3.2.

System property	Participants	Quote example
Exclusivity	P2, P6	"Is it possible for a drawing to be well recognised in both one category and another, and is it true?" (P6)
Pre-existence of categories	P2, P5	"I am very surprised, because I don't understand why it makes me a proposition when it has never seen a hat or anything else." (P2)
Optimization inertia	P1, P2, P3,	"It predicted a hat with a low confidence, and I told it "yes it is a hat", and it didn't say "ah well ok, I'm confident because you told me.""
Prior knowledge	P5, P8	"It's weird, you still get the impression that others have provided images. I feel like I'm not the first." (P5)

Table 3.2. Properties of artificial neural networks are perceived as confusing for novices along with the teaching session.

The taxonomy is composed of four properties. *Exclusivity* is the fact that each input is associated with a unique output both during train-

ing and prediction. The network cannot predict that a drawing belongs to two different categories simultaneously. P2 and P6 discussed this property since they drew ambiguous images expecting that the system would predict two categories. *Pre-existence of categories* stems from the initialization of the network with a pre-defined output size (number of categories). Thus, P2 and P5 were surprised that the model could predict a category for which no image was yet provided. *Optimization inertia* is the fact that the model is not building immediate rules from participants' demonstrations, but it optimizes parameters towards an optimum. Thus, P1, P2, and P3 were surprised that the model could still be wrong on the same image after being trained on that image. Finally, *Prior knowledge* is the fact that the model embeds prior knowledge or not. P5 wondered if the algorithm was trained with other participants' drawings beforehand: *« It's strange, you still get the impression that others have provided images. I feel like I'm not the first. »* (P5). P5 then changed her mind when the model failed on categories she had not trained yet.

P8 first believed that the algorithm relied on rules that the system designer chose. The idea of a rule-based system was primed in the questionnaire. P8 stated that *« it would be easier to provide rules rather than drawing over and over. »*. Later, P8 tried to identify the nature of these rules: *« it was part of your rules that if there's some kind of vague line, it's a wave »*. She finally intuited a notion of optimization with the idea that the rules could be adaptable to the data: *« I think it's the one that may have... not the fewest rules, but the rules that get the more easily adapted»* (P8).

Finding: About a third of the confusions expressed by the participants originate from 4 properties of neural network inherent mechanisms that we identified.

3.6 Limitations

This study does not provide a definitive picture of the large, if not infinite, space of mental models about human teaching strategies. In particular, the experimental setup largely constrains participants' behaviors. First, the interaction scenario involves sequential data creation, curation, and real-time predictions on users' sketches. This scenario limits users' apprehension of the model on data batches and exacerbates the sequencing effects identified. Second, the way participants

are prompted can constrain exploratory behaviors. If people had been prompted to generalize the model, their behavior could have been different. Ensuring that participants conceptualize the same teaching task is crucial for the reproducibility of the results. Generalizing experimental methods beyond classification tasks, such as the generation of text, image, or sound, may prove even more difficult due to the subjectivity of the task and its evaluation criteria.

IMT aims to extract teaching interaction principles independent of the learning algorithm. These interaction principles should form the IMT language. If users' understanding and strategies can be affected by the underlying algorithm (in particular, its performance), the interactions and language users can use to correct it should remain the same. The main challenge I foresee is to extract interaction principles robust to different teaching scenarios (e.g. sequence or batch) and tasks (e.g. classification, regression, generation).

3.7 Summary

We explored the way people teach learning algorithms, what strategy they use to "make it work", and what they understood from their behavior. To do so, we studied how novice users use MarcelleSketch, a sketch recognition application designed to be incrementally teachable and usable in a web browser. The application has original ML features allowing for rapid and robust training. This application has been used in both a general public online pilot workshop and individual think-aloud sessions with novice users in ML and CS.

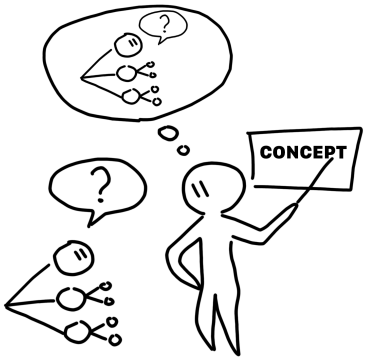
We found that participants adopted heterogeneous teaching strategies regarding sequencing and variability. The variability tends to favor the model generalization abilities, but the type of variability, and the fact it might be introduced progressively, plays a role in building an efficient classifier. We also found that repetitive sequencing at the beginning of the teaching can be detrimental to future predictions. We found that participants discovered new insights into the system by investigating transformations on existing representations. They also became aware of the importance of sequencing. Then, participants' confusions originate from four inherent properties of neural network, which fuel the discussion on the use of deep learning in IMT, discussed in 6.4.

This study shows that participants explore the limitations of the model when given sufficient room to explore. They use ambiguous or novel

examples to reason about learning behavior and system invariants. The next chapter further explores notions of uncertainty in neural networks and investigates how machine teachers perceive and use model uncertainty.

Chapter 4

Deep learning uncertainty in interactive machine teaching



This chapter takes a human-centered approach to uncertainty evaluation in deep learning. It explores how the two types of uncertainty—aleatoric and epistemic—can help non-expert users understand the strengths and weaknesses of a classifier in an interactive machine teaching (IMT) scenario. The chapter first presents a benchmark investigating uncertainty estimation using transfer learning to enable participants to retrain their model efficiently in an Interactive Machine Learning (IML) pipeline. It then investigates users' understanding of the difference between aleatoric and epistemic uncertainty and how they use uncertainty to teach an image classifier. The chapter outlines a controlled experiment in which non-experts train a classifier to recognize card images. The experiment employed a hybrid evaluation method to understand how participants perceive and use uncertainty feedback. We first tested participants' ability to predict the classifier outcome; we then conducted thematic analysis on think-aloud verbalizations and interviews. Finally, this chapter discusses Active Learning (AL) simulations in which the model is trained with a curriculum that chooses the most uncertain example at each step. The performance results inform the benefits of incorporating AL in an IMT workflow.

Contributions: I led the research conducted in this chapter and implemented the benchmark presented in subsection 4.3. Baptiste Caramiaux implemented the binary classification (in-distribution vs out-of-distribution). Pierre Thiel and I equally contributed to the experiment design presented in section 4.4 under the supervision of Baptiste Caramiaux and Wendy Mackay. Pierre and I both conducted pilot studies. Pierre Thiel programmed the interface used by participants with MARCELLE. I conducted the experiments and did both the qualitative and quantitative analysis.

4.1 Context

Although deep neural networks (DNN) have developed state-of-the-art performance on image classification problems for over a decade [Krizhevsky et al., 2012], they remain prone to predicting false positives with high confidence levels [Guo et al., 2017]. Furthermore, barely perceptible input variations can easily deceive deep neural networks [Szegedy et al., 2013]. The real-world implications of these issues are often dramatic, especially for safety-critical applications such as autonomous driving and assistive decision-making. One strategy for mitigating this problem is to estimate ML uncertainty. The research literature on ML uncertainty, particularly Deep Learning uncertainty, distinguishes between *aleatoric* and *epistemic* uncertainty. Aleatoric uncertainty captures ambiguity and noise in the data, and epistemic uncertainty captures novelty. These notions are also called *known unknown* (epistemic uncertainty) and *unknown unknown* (aleatoric) [Lakkaraju et al., Attenberg et al.]. Researchers have actively explored both aleatoric and epistemic uncertainty estimation in DNN on controlled, stereotyped data, such as Fashion MNIST [Mukhoti et al., 2021]. Within this classical ML empirical approach, uncertain examples—either ambiguous or novel—are often defined artificially for performance considerations. Especially, we lack a clear understanding of uncertainty in DNN from the user's perspective in interactive settings.

The field of Explainable AI (XAI) explores the role of uncertainty to explain ML predictions and shape people's trust in ML-based decision-making systems [Bhatt et al., 2021b, Delaney et al., 2021, Zhang et al., 2020]. Confidence levels alone can be insufficient to improve AI-assisted decision making [Zhang et al., 2020, Zhou et al., 2015]. Human-Computer Interaction (HCI) research has shown that the uncertainty inherent in probabilistic models can itself be considered as design material for interaction design [Benjamin et al., 2021]. Finally, Attenberg et al. [Attenberg et al.] developed a game-like system that encourages people to provide examples that are difficult for an ML model to classify. The authors show that people can identify more wrongly confident examples than the techniques for discovering errors in predictive models at that time (2014). Furthermore, these examples were not outliers, but coherent examples missed during model training, also called concept blindness errors.

To our knowledge, inspecting ML aleatoric and epistemic uncertainty has not been explored in the context of Interactive ML, in which participants take the role of machine teachers and iteratively train a model.

This two-levels uncertainty can theoretically help users understand if a model is incorrect because it lacks data or because the example is intrinsically ambiguous. This chapter investigates this assumption empirically through human-centered evaluations on a realistic teaching task with an IML system.

In an interactive machine teaching (IMT) context, this chapter investigates the following research questions:

- How do non-experts in Computer Science and ML use aleatoric and epistemic uncertainties when teaching a ML classifier?

- How do non-experts perceive the difference between aleatoric and epistemic uncertainty?

- Do aleatoric and epistemic uncertainties improve non-experts' understanding of the classifier and their ability to predict its outcome?

This chapter first provide the reader with the specific and fast-growing related-work of uncertainty estimation in Deep Learning. Most techniques are used offline and do not apply to HCI and fast iteration cycles on the ML model training.

I then report on the results of a benchmark study that assesses aleatoric and epistemic uncertainty estimates of real-world data using feature transfer from pre-trained models, with two different datasets. The benchmark results are used to select the appropriate method for an experiment investigating how non-experts understand both types of uncertainty. Designed for creative and educational domains [Carney et al., 2020b], this controlled experiment use the same teaching workflow as the previous user study presented in chapter 4. Participants begin with an empty image classifier that makes random predictions and then trains it incrementally by selecting and presenting a series of images. I show that teaching decisions on training set size and data variability are more critical than the type of uncertainty participants were exposed to. I also identify and discuss two ML teaching approaches adopted by participants: using uncertainty as a teaching guide or introducing systematic variations of class-dependent instances. Finally, the results also identified specific situations in which participant can identify differences between aleatoric and epistemic uncertainty.

4.2 Deep learning uncertainty estimation

The research literature on ML uncertainty, particularly Deep Learning uncertainty, distinguishes between *aleatoric* and *epistemic* uncertainty. Aleatoric uncertainty captures ambiguity and noise in the data, and epistemic uncertainty captures novelty. In this context, the concept of ambiguity refers to the gray area between the classes of a trained model. For example, if a classifier has been trained to discriminate between cats and dogs, an ambiguous example would be a picture that includes both cats and dogs. By contrast, the concept of novelty in epistemic uncertainty refers to new classes on which a model has not been trained yet. Thus in the above example of a cat-dog classifier, an image of a panda would be considered a novel instance for the model.

Before ML, the characterization of uncertainties and the manner of dealing with them was primarily the subjects of study of statisticians and engineers [Paté-Cornell, 1996, Faber, 2005, Spiegelhalter and Riesch, 2011] and largely applied to risk analysis. These fields first introduced the distinction and use of the terms epistemic and aleatoric [Hora, 1996]. Other works talk about *known unknown* or *data uncertainty* to refer to aleatoric uncertainty, and *unknown unknown*, *model uncertainty*, or *concept blindness* to refer to epistemic uncertainty [Lakkaraju et al., Attenberg et al.]. The notions spread within the Machine Learning community much more recently [Kendall and Gal, 2017]. Uncertainty estimation in Machine Learning has been explored in active learning, which aims to select the most informative instances to train the model and optimally reduce its epistemic uncertainty [Settles, 2010]. With the advent of Deep Learning and its adoption in many real-world applications, there has been an increasing endeavor in developing methods able to estimate uncertainty.

Aleatoric uncertainty captures the intrinsic randomness and ambiguity of the task and is irreducible with further training data. *Epistemic uncertainty* is caused by a lack of knowledge and is reducible given additional training data. An approach in uncertainty estimation relies on Bayesian Neural Networks (BNN) that are an extension of Neural Networks in which all parameters– weights and bias– have a probability distribution associated with them. One benefit of BNN is that they emit predictions with uncertainty i.e. the errors margin in a data point prediction.

The posterior of BNN in deep learning architectures is generally in-

tractable without strong approximations. Thus, a recent branch of research has emerged around these notions and proposed methods for approximating BNN inferences.

Aleatoric and epistemic uncertainties contributions can be retrieved in the formulation of BNN uncertainty. Indeed, Gal et al. [Gal, 2016] and Smith et al. [Smith and Gal, 2018] showed that the entropy of the predictive distribution $p(y|x, \mathcal{D})$ given a data point x and the training data $\mathcal{D}$ can be expressed as:

$$\underbrace{\mathbb{H}[Y|x, \mathcal{D}]}_{\text{entropy of } p(y|x,\mathcal{D})} = \underbrace{\mathbb{I}[Y; \omega|x, \mathcal{D}]}_{\text{epistemic}} + \underbrace{\mathbb{E}_{p(\omega|\mathcal{D})}[\mathbb{H}[Y|x, \omega]]}_{\text{aleatoric}} \tag{4.1}$$

Within the BNN approach, the distinction between aleatoric and epistemic uncertainty was first discussed by Kendall et al. [Kendall and Gal, 2017], showing neural network's limited awareness of its own confidence [Hüllermeier and Waegeman]. The research was driven by the necessity to know if additional training data can resolve uncertainty. Empirically, the ML literature showed that the challenge lies in estimating epistemic uncertainty. Estimations of the aleatoric uncertainty use well-understood measures drawn from information theory such as the Shannon entropy [Shannon, 1948]. In the following subsection, we present state-of-the-art techniques to estimate epistemic uncertainty.

Gal et al. [Gal, 2016] proposed a method to sample a trained model by randomly switching off a certain number of connections at inference (called *dropout*). Hence, one can derive N different models from a single trained model. Each model potentially provides different predictions. The variability across the N predictions of the ensemble is used as an estimator of epistemic uncertainty. Similarly, Lakshminarayanan and colleagues [Lakshminarayanan et al., 2017] proposed to independently train N DNN randomly initialized, using the same training examples. This approach is called *Deep Ensemble*. This approach also looks for disagreement among the predictions of the models' ensemble. The uncertain instances, according to the epistemic uncertainty, are those on which the ensemble strongly disagree i.e. the ensemble gives confident predictions contradicting themselves. The ambiguous instances, i.e. uncertain according to the aleatoric uncertainty, are the instances on which the models of the ensemble all give non-confident predictions. Deep Ensembles have empirically outperformed all other methods for estimating epistemic uncertainty. For example, Dropout-based techniques [Gal and Ghahramani, 2015], or techniques involving end-to-end learning of uncertainty measures [DeVries and Taylor, 2018, Franchi et al., 2020] were proved to be less successful. However,

the drawback with the Deep Ensemble approach is that training time and memory load linearly increase with the number of models in the ensemble.

Figure 4.1 illustrates both types of uncertainty in the context of Deep Ensemble. At the top, the figure depicts an ambiguous image (with respect to a handwritten digit dataset), leading to predictions with low confidence. The average confidences are low, as well as the error bars. At the bottom, the figure depicts a novel image leading to different predictions with high confidence. The average confidences remain low, but the error bars are large.

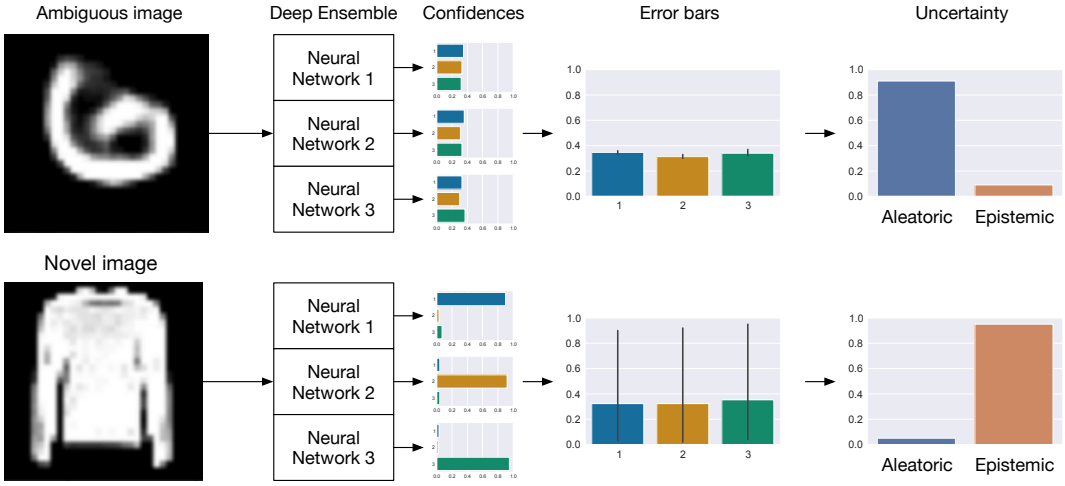


Figure 4.1. Illustration of aleatoric and epistemic uncertainties through the Deep Ensemble approach, using as input data an ambiguous image with respect to the training set made of handwritten digits recognition problem (MNIST) and a novel image (unrelated to the training set).

Recently, uncertainty estimation has been tackled through a novel approach involving the use of feature space distances and density [Lee et al., 2019, Van Amersfoort et al., 2020, Liu et al., 2020, Liu et al., Mukhoti et al., 2021]. This approach assumes epistemic uncertainty increases in sparse regions of the feature space i.e. where fewer training examples were given. This feature-based approach aims at providing a deterministic, efficient and reliable estimation of epistemic uncertainty. Postel and colleagues [Postels et al., 2020] proposed a method using the density of the feature space in different layers as a measure of the epistemic uncertainty. They found that deeper layers provide better aleatoric uncertainty while shallower layers provide better epistemic uncertainty. The challenge of this approach lies in the problem of *feature collapse* [Van Amersfoort et al., 2020], i.e. the fact that intermediate layers tend to map novel samples to the dense region of the feature space. Mukhoti and colleagues [Mukhoti et al., 2021] introduced regularization techniques of the feature space to mitigate this effect. The technique provides good results on low-resolution image datasets in which the distinction between novel and ambiguous data

is controlled and exacerbated. For example, they used MNIST as the in-distribution data and Fashion MNIST as novel data.

This short overview reveals that Deep Ensemble remains the baseline for estimating epistemic uncertainty while density-based approaches are promising to lower the computational cost of epistemic uncertainty estimation. That said, reported methods were evaluated in a setting where the training set was fixed and controlled, and the models were trained end-to-end using standard offline methods. As far as we know, epistemic and aleatoric has not been evaluated within an IML workflow, and the ML literature does not address the effect of transfer learning techniques on the DNN uncertainty estimations. We propose to explore this problem in Section 4.3 and choose adequate uncertainty estimation for the user experiment presented in section 4.4.

4.3 Benchmark Study: estimating uncertainty with transfer learning

This section explores state-of-the-art epistemic and aleatoric uncertainty estimation in a transfer learning context. The goal is to select uncertainty measures that will be used in the IMT experiment presented in the following sections.

Datasets and embeddings

We explore uncertainty estimates on two different datasets. The first dataset is derived from literature in ML. The second dataset was collected using the apparatus of the IMT experiment presented in Sections 4.2. Each dataset contains a *training set*, a *test set* and *uncertain set*. The training and test sets are comprised of In-Distribution (ID) data whereas the uncertain set is comprised of Out-of-Distribution (OoD) data. The uncertain set contains both epistemic and aleatoric instances. The datasets are:

1. The **MNIST dataset** [Lecun, Y] with additional ambiguous (Dirty-MNIST) and novel (Fashion-MNIST) images. MNIST and Dirty-MNIST are 28x28 pixel images representing handwritten digits. Dirty-MNIST includes ambiguous and noisy images. Fashion-MNIST contains 28x28 pixel images representing clothes. This dataset has been previously used in ML uncertainty assessment [Mukhoti et al., 2021]. We used 160 examples in the training set, 200 examples in the test set and 240 examples in the uncertain set.

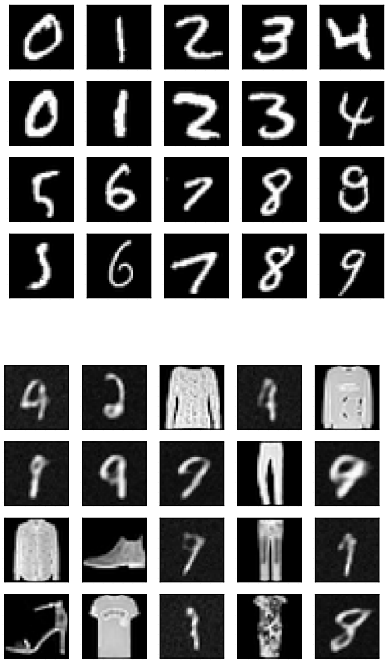


Figure 4.2: In-distribution (top) and uncertain data (bottom) for the MNIST dataset taken from Mukhoti and colleagues [Mukhoti et al., 2021]

2. The **CARDS dataset** are 350 images of playing cards we collected with a webcam fixed above a black tray used in the experiment reported in the following sections (see Figure 4.11). We collected the card images in the same lighting condition as the IMT experiment. The dataset comprises 150 training examples and 150 testing examples of the cards Nine, Queen and King, and 50 uncertain images showing both ambiguous and novel configurations. Note that the choice of the images to be added to the uncertain set was subjective. Our aim is not to create a benchmark dataset with validated labels across annotators. Rather, we designed a dataset as close as possible to the ones that participants may create in the experimental study presented in Section 4.4.

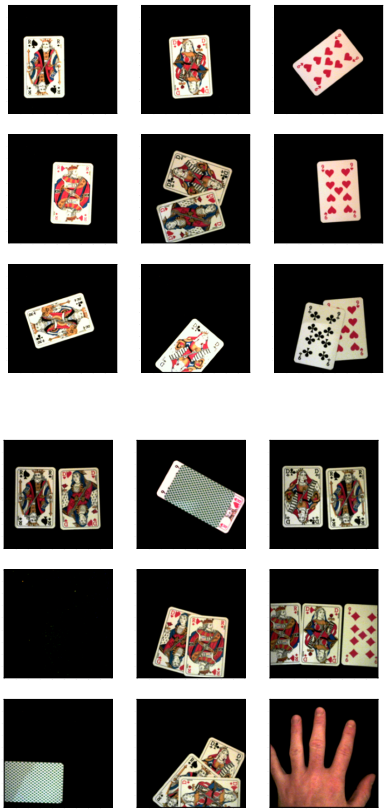


Figure 4.3: In-distribution (top) and uncertain data (bottom) for the CARDS dataset we collected.

Images from each dataset are processed through a pre-trained model and give a feature vector called embedding, on which we conduct the benchmark. This approach is standard in transfer learning, where a pre-trained model is used to create embeddings, on which a simpler classifier is trained to map embeddings values to class outputs. Transfer learning enables incremental and few-shot learning [Wang et al., 2020]. To assess the impact of the feature extraction technique on uncertainty estimation, we consider three pre-trained models available online: *MobileNetV1* [Howard et al., 2017], *MobileNetV2* [Sandler et al., 2018b] and *ResNet50* [Mukti and Biswas].

Uncertainty estimation

We present here our approach to estimate both epistemic and aleatoric uncertainty for real-time prediction in an IMT context.

Epistemic uncertainty estimation

To estimate epistemic uncertainty, we used two approaches from the related work: the Deep Ensemble baseline and a deterministic approach using Density estimation in the feature space given by the pre-trained models introduced above.

- The **Deep Ensemble** method consists of training N DNNs independently on the same training data. Each DNN in the ensemble is randomly initialized. Measuring epistemic uncertainty consists of estimating the disagreement between the predictions emitted by the ensemble, which we achieve by computing the averaged standard

deviation of the per-class likelihoods:

$$u(z) = \frac{1}{N}\sum_{i=1}^{N} \texttt{std}([p_i^k(z)]_{k=1..M}) \tag{4.2}$$

where $p_i^k(z)$ is the probability of class i given by the k^{th} model in the ensemble, for input data z, `std` is the standard deviation computed over the models in the ensemble. In this paper we consider an ensemble of 3 Multi-Layer Perceptrons (MLP) with two hidden layers of 64 and 32 neurons. Each MLP is placed on top the pre-trained model. During training, only the MLPs are trained.

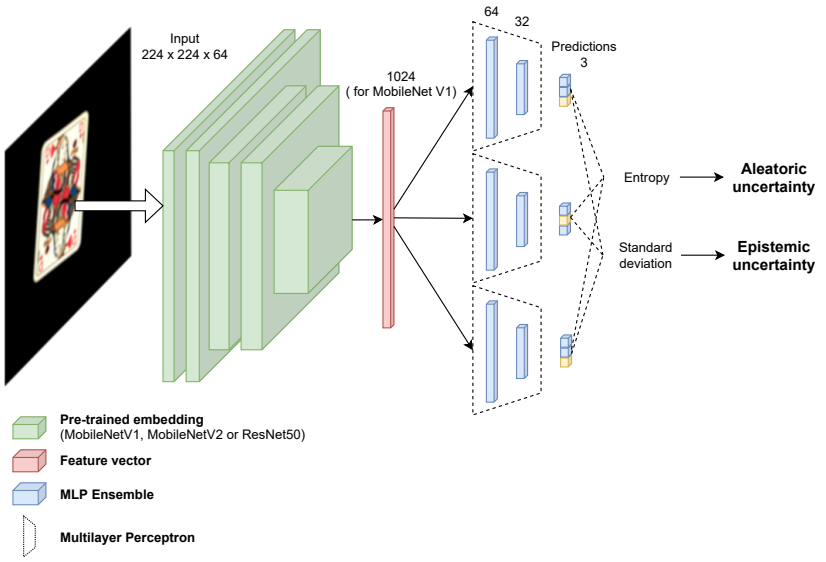


Figure 4.4. Schema of the Deep Ensemble approach for calculating epistemic and aleatoric uncertainties.

- The **Density estimation** computes the data density in the feature space as created by the pre-trained models. Novel images are assumed to be far from the dense area composed of the training data projected in the feature space. They will therefore obtain low likelihood probability under the density model. The density-based uncertainty measure relies solely on data representation in the feature space and does not require the training of a classifier. We use two different approaches:

  1. Gaussian Mixture Model (GMM): each Gaussian component is centered on a class from the training set. The model learns the variances and the mixing weights. Epistemic uncertainty is estimated using the weighted log-likelihood of a new input data point under the trained GMM.

2. Gaussian Density: one density function using Gaussian kernel is trained per class on data embeddings created by the pre-trained models. Measuring epistemic uncertainty is performed by computing the sum log-likelihood over the density models.

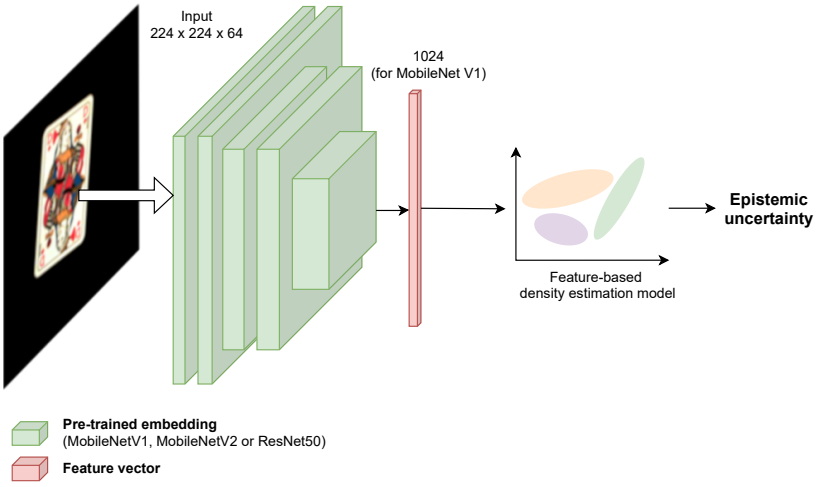


Figure 4.5. Schema of the feature-based approach for calculating epistemic uncertainty.

Aleatoric uncertainty estimation

To estimate aleatoric uncertainty, we follow the standard approach by computing the entropy of the softmax distribution provided at the output of the classifier [Mukhoti et al., 2021].

The entropy computed on the softmax probability distribution is as follows:

$$H(z) = -\sum_{i=1}^{N} p_i(z) \log_2 p_i(z) \tag{4.3}$$

where z is an input data point and $p_i(z)$ is the softmax value for class i. We note that the uncertainty is calculated downstream from the predictions' probability emitted by the Neural Network.

The pre-trained embeddings, models and acquisition functions used in the benchmark are summarized in table 4.1

Results

We assessed uncertainty estimates through their performance in detecting uncertain data (out-of-distribution) from test data (in-distribution). We consider the problem as a binary classification between positives (test data) and negatives (uncertain data) and use the area under the ROC curve (AUROC) as the performance metric. We also report a

Type of uncertainty	Embeddings	Model	Acquisition function
Aleatoric uncertainty	MobileNetV1 MobileNetV2 ResNet50	MLP Ensemble	Shannon Entropy
	MobileNetV1 MobileNetV2 ResNet50	Single MLP	Shannon Entropy
Epistemic uncertainty	MobileNetV1	MLP Ensemble	Standard deviation
		GMM	Log-lokelihood
		Gaussian Kernel	Density estimation
	MobileNetV2	MLP Ensemble	Standard deviation
		GMM	Log-lokelihood
		Gaussian Kernel	Density estimation
	ResNet50	MLP Ensemble	Standard deviation
		GMM	Log-lokelihood
		Gaussian Kernel	Density estimation

Table 4.1. Summary of the approaches used in the benchmark. Each techniques was applied on the *MNIST dataset* and the *CARDS dataset*.

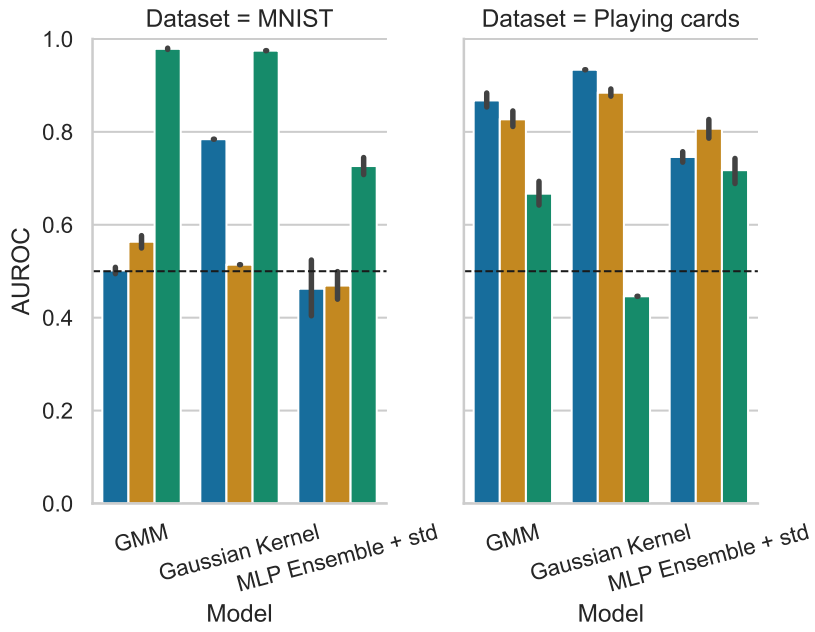
complementary analysis on the influence of pre-trained models on uncertainty estimation.

Epistemic uncertainty estimation

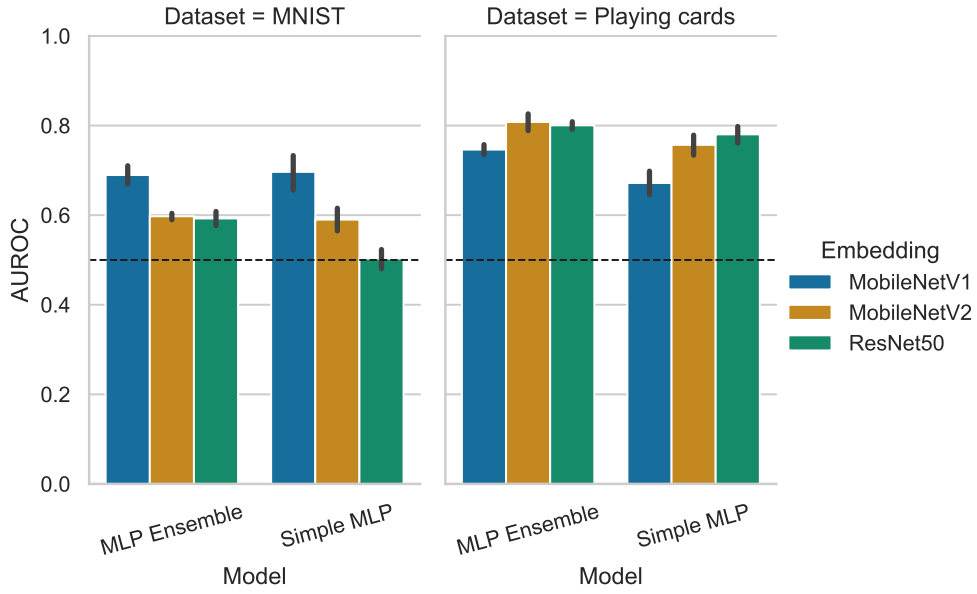
Figure 4.6.a reports the results obtained considering epistemic uncertainty measures. The results showed an influence of the type of embedding (MobileNetV1, MobileNetV2, or ResNet50) on the detection performance. On the MNIST dataset, techniques using ResNet50 performed significantly better than when using the two other embeddings. In addition, combining with density-based approaches provided nearly optimal detection rates (AUROC=0.98 for both GMM and Gaussian density). On the CARDS dataset, both MobileNetV1 and MobileNetV2 achieved higher performance than ResNet50. Combining with density-based approaches also showed higher performance (AUROC=0.93 [resp. 0.87] for Gaussian density [resp. GMM]). Hence, this result showed that epistemic uncertainty on the playing card data is better estimated using MobileNetV1 as an embedding and Gaussian Kernel density.

Aleatoric uncertainty estimation

Figure 4.6.b reports the results obtained considering aleatoric uncertainty measures. On the MNIST dataset, we found that a MobileNetV1 embedding yields the highest AUROC measure, regardless of whether there is an MLP or an ensemble of MLPs used to produce the prediction likelihoods (AUROC=0.69 [resp. 0.76] for MLP Ensemble [resp. Simple MLP]). On the CARDS dataset, we found fewer differences between embedding and techniques. The highest detection rates are about 0.8.



(a) Epistemic uncertainty estimation



(b) Aleatoric uncertainty estimations

Figure 4.6. (a) AUROC metric of a binary classifier detecting uncertain data from in-distribution data with the epistemic uncertainty estimation techniques, considering different datasets (MNIST and CARDS) and embeddings (MobileNetV1, MobileNetV2 and ResNet50). (b) AUROC metric of a binary classifier detecting uncertain data from in-distribution data with the aleatoric uncertainty estimation techniques and considering different datasets (MNIST and CARDS) and embeddings (MobileNetV1, MobileNetV2 and ResNet50). The dashed black line represents random sample assignment between uncertain and in-distribution.

Detecting ambiguous and novel data in the play card dataset

We focused on the CARDS dataset. We inspected the distribution of uncertainty estimates for in-distribution, ambiguous and novel data. We used the best techniques from 3.3.1 and 3.3.2: Gaussian Kernel on MobilenetV1 for estimating epistemic uncertainty, and MLP Ensemble on MobilenetV1 for estimating aleatoric uncertainty. Figure 4.7 reports

the histograms: the left panel reports the histogram of epistemic uncertainty estimations (Gaussian Kernel), the right panel reports the histogram of aleatoric uncertainty estimations (Deep Ensemble). Both techniques use the MobileNetV1 embedding.

Novel data has high values from the Gaussian Kernel density estimation. By contrast, novel data have low entropy values computed on the MLP Ensemble probability distributions and are confused by positive data. Ambiguous data, however, has intermediate entropy values.

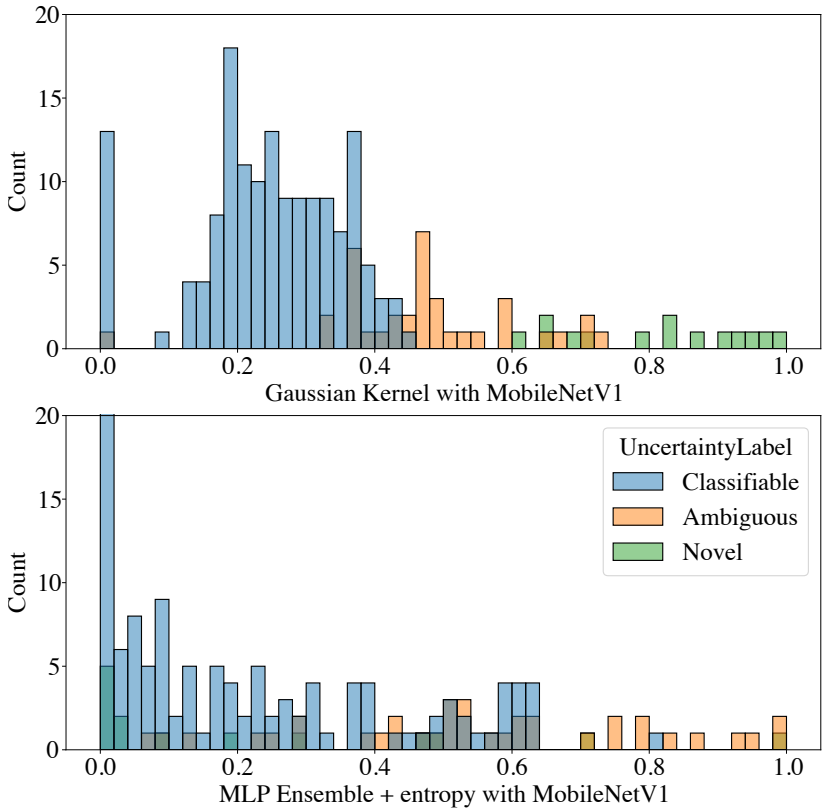
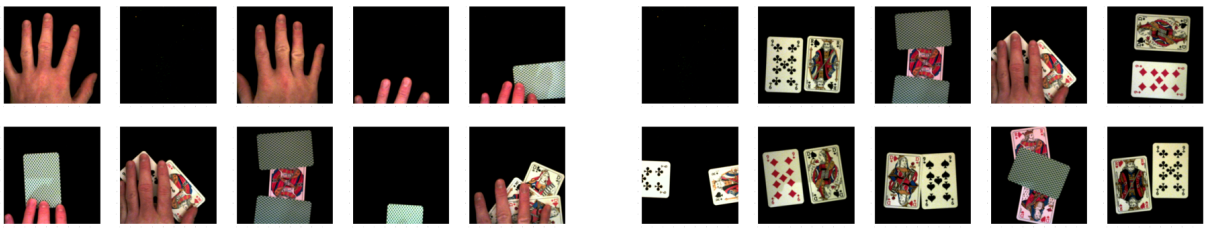


Figure 4.7. Distribution of the playing card data according to (left) the epistemic uncertainty (Gaussian Kernel on MobileNetV1 features) and (right) the aleatoric uncertainty (entropy on MLP Ensemble using MobileNetV1 features). The label "classifiable" refers to data from the test set. Ambiguous and novel labels have been assigned to instances from the uncertain set by the first author.



(a) Epistemic uncertainty estimation **(b) Aleatoric uncertainty estimations**

To help the reader appreciate the data detected as uncertain, Figure 4.8 depicts the images located at the highest values of both uncertainty

Figure 4.8. Images from the playing card dataset that obtained extreme values according to the two types of uncertainty: (a) Gaussian Kernel with MobileNetV1 for the epistemic uncertainty and (b) entropy on a MLP Ensemble using MobileNetV1 features for the aleatoric uncertainty.

measures. We observed that high estimates of epistemic uncertainty showed out-of-distribution data, where the background may be dark or showing a hand. This data can be considered as novel in the sense that the *concept* defined by a hand or a dark background is novel for playing cards. On the other side, high estimates of aleatoric uncertainty show ambiguous images, in which two cards are shown instead of one.

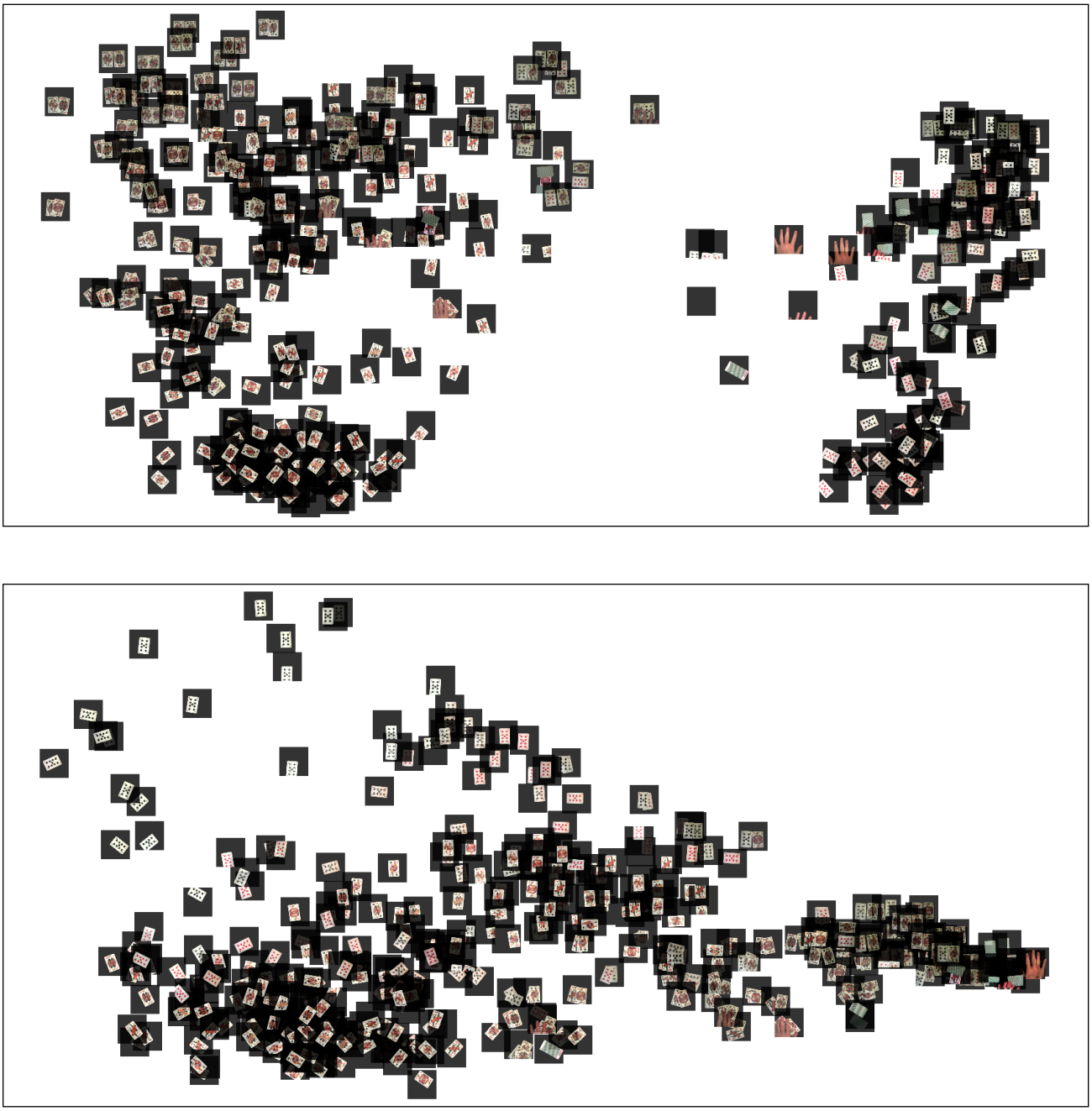
Analysis of variance

Finally, we report further analysis to understand pre-trained model's influence on detection performance in epistemic uncertainty. More precisely, we inspect whether the distribution of variance within the space influences the detection performance. We performed a Principal Component Analysis (PCA) on the training set through each pre-trained model— MobileNetV1, MobileNetV2 and ResNet50. We kept the 10 first principal components and computed the variance explained by each component. Finally, we computed the entropy of these 10-dimension vectors. High entropy means that the variance is spread over the components, while low entropy means that the variance is concentrated on fewer components. Table 4.2 reports the entropy values together with the averaged AUROC values across models. It shows that entropy is intrinsically linked to detection performance: higher entropy values imply better detection. In other words, having an embedding where the variance is spread over a higher number of components increases the detection capacity of epistemic uncertainty estimates.

	MNIST		**Playing cards**	
	entropy	mean(AUROC)	entropy	mean(AUROC)
MobileNetV1	1.72	0.63	1.98	0.87
MobileNetV2	1.69	0.51	1.98	0.84
ResNet50	1.99	0.89	1.87	0.59

Table 4.2. Entropy of the ten first components of the PCA on both datasets and the three different embeddings.

In figure 4.10, we display the dataset according to the first two components to illustrate the how data set spread over two first components.



Findings: Feature-based uncertainty estimation using pre-trained embeddings can be used to identify uncertain examples, both aleatoric or epistemic and on standard data (MNIST) or specific data (CARDS). The variance of the data among the pre-trained embedding is an indicator of the ability of model ability to distinguishing uncertain examples.

Figure 4.10. The CARDS dataset across the first two principal components using ResNet5

Limitations

On the difficulty of providing annotations of uncertain instances

Researchers usually rely on annotated data that distinguish ambiguous from novel instances to evaluate both uncertainty estimates (aleatoric and epistemic). However, it might be very challenging for users and researchers to make a clear distinction between ambiguous and novel data in real-world problems. ML researchers working on uncertainty estimation typically use stereotyped datasets that clearly define ambiguous and novel data. For example, Mukhoti et al. [2021] used handwritten digits as in-distribution data but clothing items as novel data. Such distinction might sound arbitrary in a real-world problem. We typically encountered this problem when labeling the CARDS dataset. Differentiating between ambiguous and novel examples was not a trivial task and might be subjective.

On the technological dependency on pre-trained model

We saw that the choice of a pre-trained DNN is crucial when using real-time uncertainty estimation in a transfer learning setting. We showed that the variance distribution of the data in the feature space dimensions influences the participants' ability to detect uncertain examples (ambiguous and novel) from in-distribution examples. Existing approaches retrain the embedding using regularization techniques for ensuring sensitivity and smoothness of the feature space [Mukhoti et al., 2021, Van Amersfoort et al., 2020]. In the context of IML, where iteration cycles are tight [**?**], we could not afford to retrain the whole model generating the embedding. However, we assume that an embedding extractor calibrated for the task can be trained offline. Then, one could freeze its parameters for real-time uncertainty estimation.

The main problem is that it introduces a task-dependent technological dependency for uncertainty estimation. Rather than using out-of-the-box parameters from Imagenet, we could develop pre-trained embedding for large recognition tasks (e.g. medical images, written characters, etc.), ensuring accurate uncertainty estimations in related tasks. With this approach, we encourage further research to understand Transfer Learning for uncertainty estimation and provide pre-trained embedding enabling interactive machine teaching.

4.4 Experimental study

The benchmark study looked at aleatoric and epistemic uncertainty estimates on two fixed data sets in order to identify appropriate uncertainty measures for an interactive image classification task. Here, our focus shifts to the human teachers: we are interested in the strategies that novices use to predict the behavior of the classifier, given the two types of uncertainty. We conducted a one-factor within-participant experiment where participants interactively teach an image classifier to recognize three types of ordinary playing cards—nines, queens, and kings—. The two conditions are the type of uncertainty used as feedback: aleatoric or epistemic.

Participants

We recruited 16 participants (11 women, 5 men, 15 aged between 18 and 29, 1 above 30). We recruited participants using mailing lists and social networks from the university, associated schools, and student residences. We selected participants with little or no computer science training. They are from biology (6), design (4), sociology (1, former student), philosophy (1), linguistics (1, former student), math (1), economics (1) and chemistry (1). Half have never programmed, 6 have minimal programming training, 2 have programming experience, but not as their main activities. Six participants had never heard of Machine Learning. The rest have heard about it through the media but have never had any theoretical or practical training. Participants received 10 euros in compensation.

Setup

Apparatus: Figure 4.11:*(top)* shows the setup, which includes a 42" monitor and a mouse for interacting with the experiment application and a camera stand with a fixed Logitech C270 HD webcam located 25 cm above a tray covered with black fabric, where participants place cards to be trained or tested. Participants have a set of 12 playing cards (4 nines, 4 queens and 4 kings from each suit) from a standard French deck with the Paris pattern. This deck represents the classes that participants must teach to the classifier. They also have access to the rest of the deck, blank sheets of paper, a pen, and a black and a red marker.

Software[1]**:** Figure 4.11:*(bottom)* shows the experiment application, developed using the Marcelle [Françoise et al., 2021] interactive machine learning (IML) toolkit for building interactive web interfaces based

[1] The source code of both the Marcelle application and the benchmark presented in section 4.3 is available at https://github.com/teo-sanchez/teaching-uncertainties-iui2022.

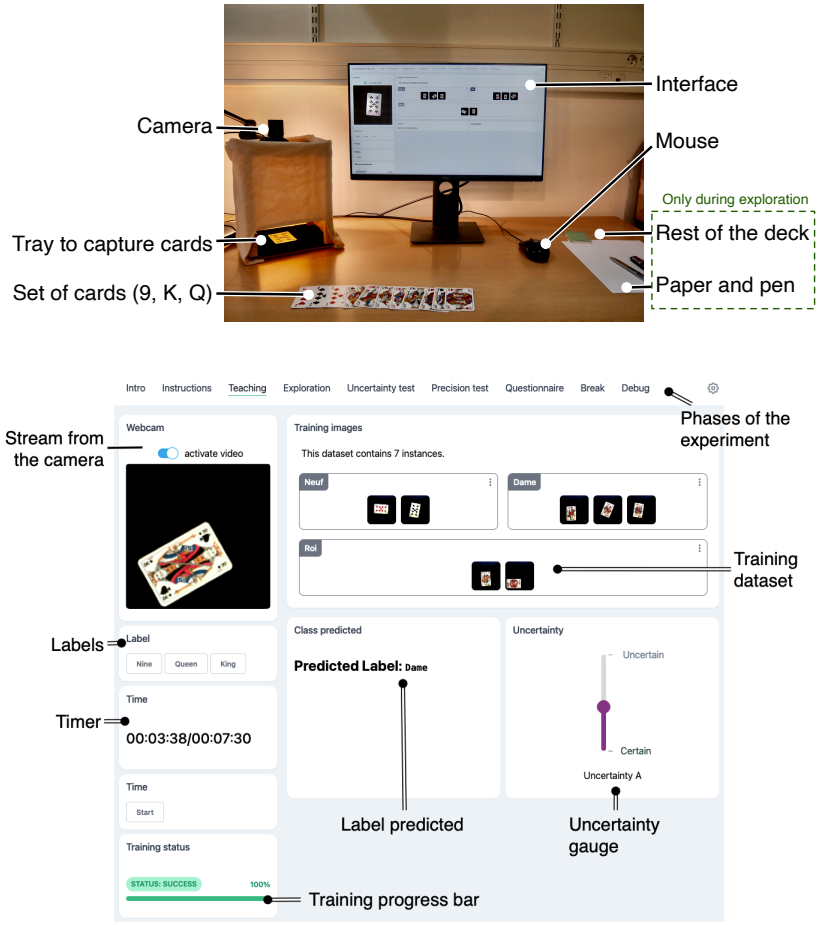


Figure 4.11. *Top:* **The setup includes a screen, mouse, and camera stand for recording individual cards. Participants have access to the 12 training cards, the rest of the deck, paper and pens.** *Bottom:* **The application displays the live webcam feed, training set, prediction, uncertainty estimation and a series of tabs associated with each step of the experiment. The above interface is not shown in full screen for legibility. During the experiment, the different components are arranged in the same way but in full screen format.**

upon ML pipelines. The application and the model training and inference all run in JavaScript. The application also uses a python server to run a python script that performs Gaussian Kernel density estimation with each new data input. We use a NeDB backend for data storage. The software displays 9 tabs. The first seven describe the successive steps of the experiment (see Section 4.4): *Introduction*, *Instructions*, *Teaching*, *Exploration*, *Uncertainty Test*, *Classification Test*, *Questionnaire* and *Pause*. The final tab *Debug* is for us to retrain the classifier in case the application crashed, which did not happen during the experiments.

Machine Learning pipeline and uncertainty estimation: Figure 4.12 summarized the choice made during the benchmark on the ML pipeline and uncertainty estimation techniques. We use a pre-trained MobileNetV1 model to process the input image. The MobileNetV1 output (features) is used as input to both the 3 MLPs (2 layers of 64 and 32 hidden units) and the density estimation algorithm. Aleatoric uncer-

tainty is computed using the entropy on the MLPs' outputs. Epistemic uncertainty is computed through the Gaussian kernel density model.

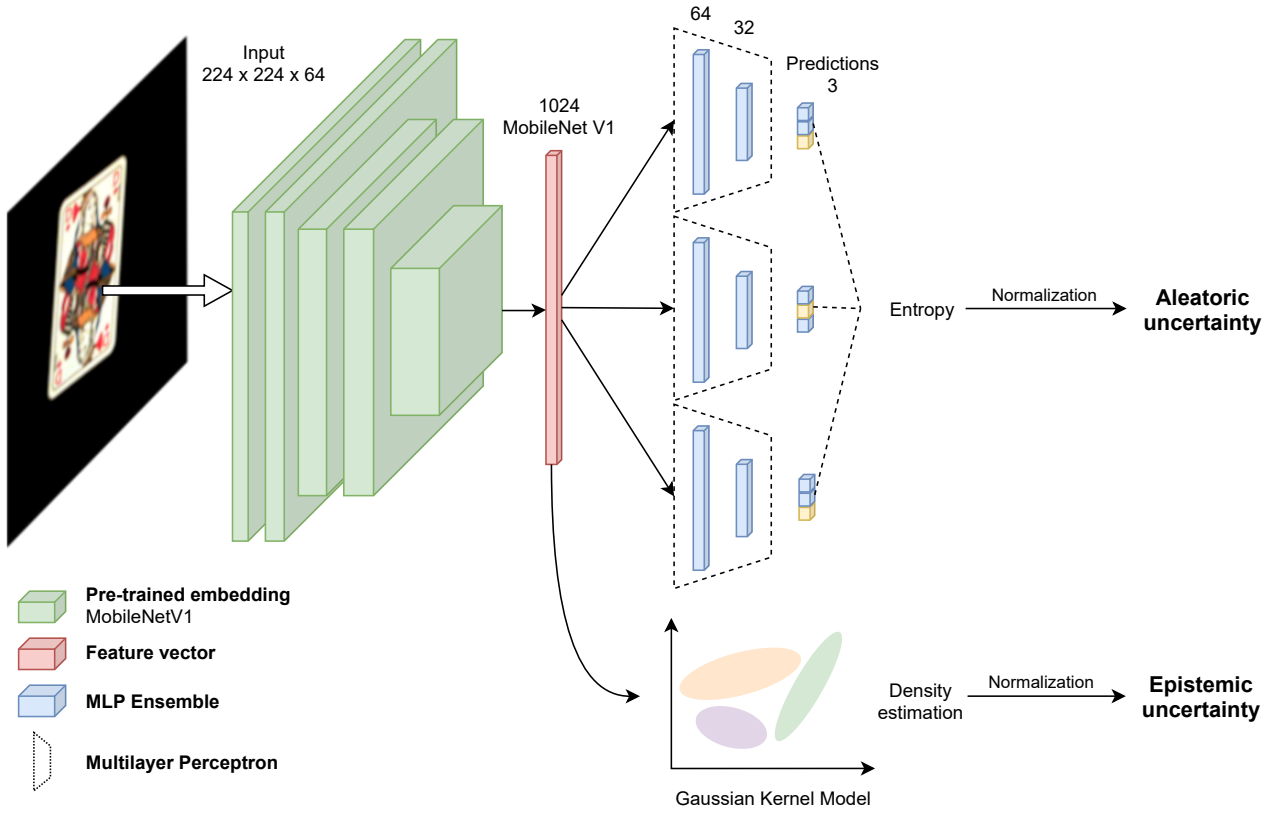


Figure 4.12. Machine Learning pipeline and uncertainty estimation chosen for the user experiment. The first image represent a frame from the video stream. All the computation are performed in real-time. The prediction given on the interface is the averaged prediction over the MLP Ensemble

Procedure

We use a one-factor, within-participants design with two conditions: aleatoric and epistemic uncertainty. Participants first watch an introductory video that describes the purpose of the study, a short primer on machine learning, a description of the setup, and the procedure steps. We ask participants to read and sign a consent form. Next, participants watch a video introducing ML uncertainty concepts, the experiment interface, and the basic training task. We label the two uncertainty measures A and B. **Uncertainty A corresponds to aleatoric uncertainty, uncertainty B corresponds to epistemic uncertainty**. We only tell participants that they correspond to two different methods for computing uncertainty. Participants do not know what they are nor how they are computed. Participants then complete two iterations of the following five steps, one for each uncertainty condition, counterbalanced for order across participants.

1. **Teaching**: Participants have 7.5 minutes to provide the classifier with a series of training examples. The participant first picks a card and places it on the tray. The participant clicks on a label —nine, queen or king— to add a new labeled image to the training set. After the participant labeled three examples, the model is trained for the first time and the timer starts. Until then, the system gives real-time predictions from the camera video stream. The video frames are used to predict both the label and the uncertainty. The name of the predicted label is displayed while the uncertainty is represented by a gauge (high values correspond to high uncertainty). Each time the participant labels a new image, it launches training on the updated training set again. We asked participants to provide a verbal comment to explain their actions, their current understanding of the classifier's behavior, and any confusion about the classifier or the uncertainty measures.

2. **Exploration**: The aim of this phase is for the participant to understand how the classifier behaves. We do not allow the participant to label new images. Therefore, the classifier is not further trained on new examples. However, the participant can continue placing cards under the camera to explore the classifier predictions. They can use kings, queens, nines, or any other or use cards from the remaining deck. As before, participants provide a verbal comment as they work. They can also write notes to help them memorize the classifier behavior.

3. **Uncertainty test**: Participants see a sequence of 12 new card images on the interface. For each card, participants use a slider to manually set the level of uncertainty that they predict the trained classifier would display for this card. 7 out of 12 cards are in-distribution i.e. they represent either a nine, a queen or a king. 5 out of 12 are out-of-distribution. They represent an empty image (1), a hand (1), and two different cards on the same image (3). None of the 12 images are cards from the rest of the deck.

4. **Classification test**: Participants see a sequence of 20 new images of playing cards among nines, kings, and queens. Participants must predict if the system will successfully classify them or not. Participants receive one point for each correct prediction: either by correctly predicting that it will succeed or by correctly predicting that it will fail. They lose a point for each incorrect prediction and neither gain nor lose a point if they answer that they do not know.

5. **Questionnaire**: Participants answer five questions about the teach-

ing session and their perception of the uncertainty measure using 5-point Likert scales. One question is about the classifier's performance; the next three questions are about the usefulness of the uncertainty measure to identify the examples that the classifier knows, does not know, or is ambiguous about. The last question is about the predictability of the uncertainty measure. The questionnaire is given in appendix.

6. **Interview**: The experiment ends with a semi-structured interview based on the participants' questionnaire answers. It also comprises open-ended questions about their comments during the teaching, exploration and test steps, and how they describe the system's uncertainty behavior.

7. **Pause**: After the first condition, participants take a short break before starting the second condition.

After completing the above five steps for each of the two uncertainty conditions, participants complete a questionnaire with demographic information, their background level of knowledge of programming and understanding of machine learning, their reasons for participating in the study, and their level of engagement with the tasks in the experiment.

Data collection and analysis

We collected all the images used for training by each participant, the weights of the model trained after each example, and the participants' answers given during the uncertainty test, classification test, and questionnaire. We also recorded audio during all steps and video during the exploration step. To preserve anonymity, we transcribed the audio of participants' verbal comments throughout the experiment and conducted a mixed thematic analysis [Braun and Clarke, 2006] with anonymized transcripts. We first identified themes that emerged from analyzing the transcripts from the first eight participants; and then examined the transcripts of the remaining eight participants according to those themes. We iterated on the themes by re-examining all 16 participants.

We divided the themes in two groups:

1. *Teaching curricula* contains four themes: systematic, non-systematic, exhaustive and exclusive curricula.

2. *Understanding of the uncertainty measures* contains four themes: explanations, differences, usefulness and confusions.

We also present the results of the Likert-scale questionnaire to support for the qualitative results. Regarding the quantitative analysis, we computed the following measures to be compared between the two conditions:

- *Participant uncertainty test score*, calculated the average proximity between the uncertainty values chosen by the participants on the 12 images and the actual uncertainty estimation for the condition, either *aleatoric uncertainty* (A) or *epistemic uncertainty* (B). To have a performance score that increases when participants succeed, we calculate the proximity as one minus the average distance between participants' response and the actual value:

$$score_{uncert} = 1 - \frac{1}{N}\sum_{i=1}^{N} |u_{model}(X) - u_{guessed}(X)| \tag{4.4}$$

  with $u_{model}(X)$ being the actual uncertainty on the image X displayed and $u_{guessed}(X)$ the uncertainty estimated by the participant for the same image X during the study. In our case, N equals 12, the number of images tested.

- *Participant classification test score*, calculated as described in subsection 4.4 i.e. the number of times participants correctly predicted the classifier outcomes minus the number of wrongly predicted classifier outcomes.

- *Classifier accuracy*, calculated as the number of times the classifier found the correct label among the test images, divided by the number of test images (20).

- *Number of training examples* is a simple counting of the number of images given by the participants to the classifier.

- *Training set variability*, computed within a class using Euclidean distance between pairs of images in the feature space, i.e. between the output vectors of MobileNetV1 for each drawing. We only calculate the similarity between images of a same a class. It does not make sense to compute a similarity between images from different concept class. Finally, we averaged the computed distances between all pairwise combinations of instances within a class. We then averaged

the per-class variability for each participant. Formally:

$$V_{training_set} = \frac{1}{3} \sum_{c \in classes} \frac{1}{C^2_{size(c)}} \sum_{X_i, X_j \in c} d(M(X_i), M(X_j)) \tag{4.5}$$

with $C^2_{size(c)}$ the number of combinations of 2 instances in the class c, d the Euclidean distance, and $M(X)$ the feature vector after passing the input image X through the MobileNet network. To help the reader appreciate the variability across participants, Figure 4.13 and Figure 4.14 depict the training sets of the most variable and least variable teaching sessions.

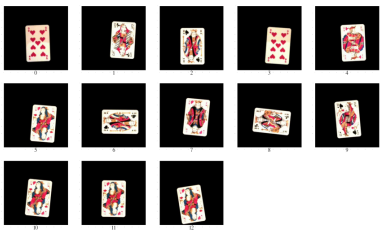


Figure 4.13: The least variable training set (participant 15) from the teaching sessions of the participants.

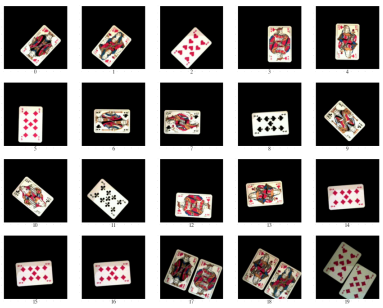


Figure 4.14: The most variable training set (participant 4) from the teaching sessions of the participants.

4.5 Results

This section reports results on (1) participants' ability to predict the behavior of the classifier and (2) their ability to explain how it behaves. The results on (1) are presented is section 4.5, and studied through the quantitative analysis of the **uncertainty test** and **classification test** scores introduced in section 4.4. The results on (2) are presented in section 4.5 through the analysis of the think-aloud verbalizations from the *teaching* and *exploration* phases and from the **interviews** conducted in each condition. On average, participants managed to train their classifier with a mean classification accuracy of 0.83 (std = 0.09).

Ability to predict the classifier behavior

After teaching, participants successfully predicted the classifier outcomes during the test phase. One-way ANOVAs reveal that participants predicted both model classification and uncertainty above chance (F = 96 and $p_{value} < 0.001$ for classification test and F = 25 and $p_{value} < 0.001$ for uncertainty test).

Influence of the type of uncertainty

We inspect whether the type of uncertainty affects participants' ability to predict the classifier behavior. More precisely, we test whether this factor influences both the participants' uncertainty test and classification test scores.

When grouping teaching sessions across participants according to the two conditions, *aleatoric uncertainty* and *epistemic uncertainty*, two one-way ANOVAs reveal that the type of uncertainty has no significant effect on participants' uncertainty test score (F = 0.43 and p_{value} = 0.52) nor on classification test score (F = 0.135 and p_{value} = 0.72). This sug-

gests that participants do not predict one type of uncertainty better than the other after teaching the classifier. Moreover, it indicates that the type of uncertainty shown has little effect on participants' ability to predict the classifier outcomes, both for classification and uncertainty. The Likert scale questionnaire suggests that the uncertainty predictability is subject to a great variability across participants and does not depend on the type of uncertainty used, as depicted in Figure 4.15. Finally, we performed a similar test using classifier accuracy as an independent measure. We also found no significant effect of the type of uncertainty (F = 0 and p_{value} = 1.0).

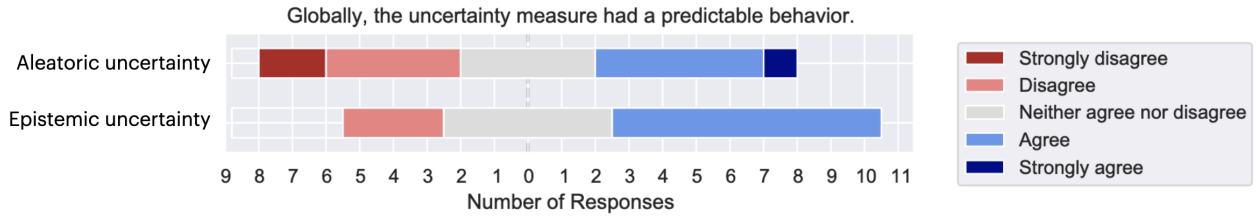


Figure 4.15. Answers to the question "Globally, the uncertainty measure had a predictable behavior" exhibit a great variability across participants no matter the type of uncertainty shown as feedback.

Order and learning effect

Participants, especially novices, might be subject to a learning effect: their ability to perform the task increases from the first iteration to the second. We found that participants gave more variable images in the second iteration than in the first one. A Student's t-test shows that the training set variability (see section 4.4) is significantly higher in the second iteration than in the first one (F = 13.4 and p_{value} = 0.001). We can explain this observation by the fact that participants usually explore the level of variability the classifier can handle in the first iteration. Thus, we assume that participants gave more variable images in the second iteration because they already explored the limits of the classifier in the first iteration.

Furthermore, participants better estimate uncertainty after the second iteration, independently of the type of uncertainty. A t-test shows a borderline effect of iteration on the participant uncertainty test score (F = 3.24 and p_{value} = 0.081). However, the iteration does not help in estimating the classification behavior.

One participant commented on this learning effect: *«I don't know if it's the lessons I learned from the other one that made me behave this way for this one or if it's because the measure of uncertainty is different and therefore it induced a different behavior in me. I really can't say.»* (P2).

Accuracy

We found that the classifier accuracy is positively correlated with the participant classification test score (R = 0.60 and $p_{value} < 0.001$) but not with the participant uncertainty test score (R = 0.27 and p_{value} = 0.13). This result shows that it is easier to estimate the classification accuracy when the model is well-trained, probably because participants do not have to remember all the cases where the classifier might fail. However, it is worth noting participants' ability to predict their classifier uncertainty is not influenced by the classifier accuracy.

Number of training examples and variability

The variability in the test results suggests that the individual specificity of the teaching prevails over the effect of the type of uncertainty. We propose looking at the teaching curriculum i.e. the strategy of organizing the training examples and introducing complexity. In these quantitative results, we focus on two characteristics of the teaching curricula: the training set size and variability (described in section 4.4).

First, we found that participants who gave more training examples also gave more variable ones. Indeed, we found a positive Pearson's correlation between the size of the training set and the training set variability (R = 0.50, $p_{value} < 0.01$). We also found that participants who give a higher number and more variable training examples train more accurate classifiers. The size and variability of the training set are both correlated with the classifier accuracy (R = 0.50, $p_{value} < 0.01$ for the training size and R = 0.57 and $p_{value} < 0.001$ for the variability). In the same way, the size and variability of the training set are also both correlated with the participant accuracy test score (R = 0.62, $p_{value} < 0.001$ for the training size and R = 0.47 and $p_{value} < 0.001$ for the variability). Bigger and more variable training sets produce a more accurate classifier, and the outcomes of an accurate classifier are easier predict for participants.

More importantly, we found that only the variability of the training set correlates with high scores in the participant uncertainty test (R = 0.40 and p_{values} = 0.024). We assume that exploring more variable configurations might trigger greater variations in uncertainty between these configurations, which in turn would help participants understand the uncertainty dynamics. Finally, neither the size of the training set nor the classifier accuracy affects the participants' ability to predict the classifier uncertainty. We report these results in Figure 4.16, as well the linear regressions between the size and variability of the training

set and the participants' classification and uncertainty scores.

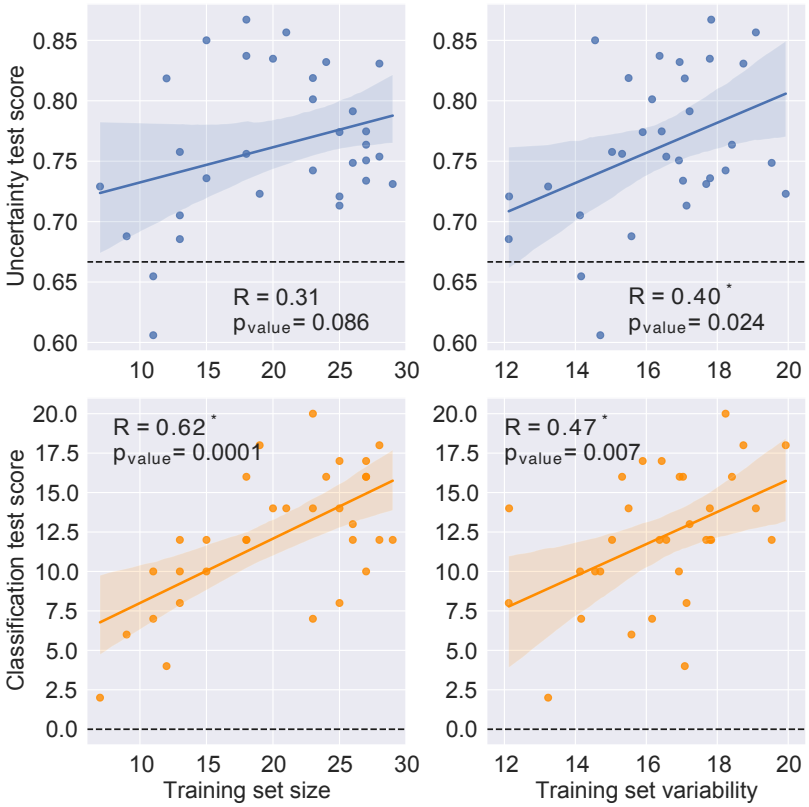
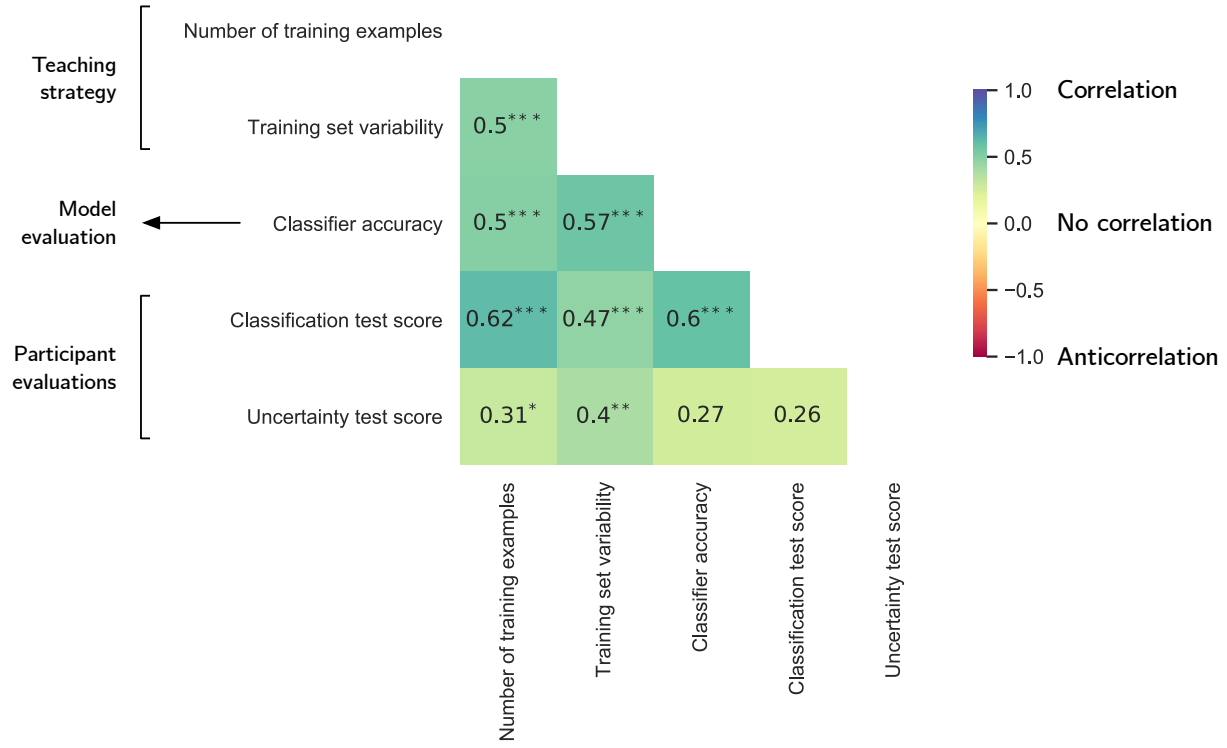


Figure 4.16. Linear regressions between the training set size (the first two) and variability (the last two) and the participants' uncertainty test score (blue) and classification test score (orange). The dashed black lines represent the chance baseline i.e. random responses during the test phases.

In Figure 4.17, we summarize the significant correlations found in subsections 4.5 and 4.5.

The findings of this subsection 4.5 can be summarized as follow:



Findings:

- Participants can successfully predict the classifier outcomes both in term of predictions and uncertainty. Participants' ability to predict the system behavior does not depend on the type of uncertainty shown.
- The choices made during teaching about the number of training examples and their variability affect the participants' ability to predict the classifier's behavior. The training set size only improves participants' ability to predict the classifications made by their classifier. The training set variability improves both participants' ability to predict their model classification and uncertainty.
- The more accurate the classifier, the more easily participants can predict the classification made by their classifier—however, this correlation does not hold when predicting the type of uncertainty.

Figure 4.17. Correlation matrix between characteristics of participants' teaching strategy, the resulting classifier and the participants' evaluation on their ability to predict the classifier outcomes (predictions and uncertainty). All correlations are positive and significant correlations are indicated with stars as follow: $*$ for $p_{value} < 0.1$, $**$ for $p_{value} < 0.05$ and $**$ for $p_{value} < 0.001$.

Participants' teaching curricula

This section examines the participants' teaching curricula in more detail than the two characteristics —size and variability— introduced in the previous section. We analyze participants' verbalizations to categorize and describe the different teaching curricula employed and how these curricula relate to the uncertainty. We use acronyms to quote participant number and the condition in which the quote was verbalized. For example, P3-A refers to condition A (aleatoric uncertainty) of participant 3.

Uncertainty as a guide

Four participants—P2-A, P7-B, P9-B, P8-B and P15—AB used the uncertainty measure as a guide to look for uncertain images and add them to the training set. They expect this strategy to optimally reduce the epistemic uncertainty and errors: *« The greater the uncertainty, the more careful I am. It's more the negative that makes you adjust than the positive. [...] We are more driven to fix what's wrong than to take care of what's right. Actually that's it, I have to test it by moving it around, to see what it does in terms of uncertainty. »* (P2). Similarly, P9 explicitly looked for the most uncertain region and validated the class. P8 also had a spatial metaphor to describe this strategy: *«I tell myself that I just have to train it as much as possible when it is the most uncertain, so that he can fill the void it has.»* (P8).

These strategies echo the Active Learning paradigm [Settles, 2010] where a model tries to select the most uncertain—therefore informative— instances in order to improve performance while reducing the amount of data resource.

Systematic teaching curricula

Participants can adopt systematic teaching curricula. Systematic curricula imply a planned order in which images are added, usually by series of colors or inclination across all classes. These strategies are usually conducted after participants realize that imbalanced variations across classes cause misclassifications.

Participants 4-B, 6-AB, 7-B and 8-A were explicitly systematic in their curriculum. For example, P4 said *« I did all the same series in one direction, the 9 of diamonds, the queen of diamonds and the king of diamonds, all in the same order each time. [...] It's already obvious that it's better trained than the first time, I think I dispersed it a bit too much the first time and the fact to be ordered right away, it doesn't get lost and it concentrates on the essentials of the cards»* (P8). Among the participants mentioned, two

claimed that being systematic helped them understand the uncertainty behavior. Participant 8 said: *« The fact that I created a protocol allowed me to understand better how the gauge reacts. I trained all the cards the same, with the same number of images, four red, four black, four different angles. It's like I trained it in a more neutral way. This way, I understand its behavior a bit more than the first time.»* (P8).

The teaching sessions in which participants claimed to use a systematic curriculum have significantly larger training set size and variability than others according to Student's t-tests (F = 4.16, p_{value} = 0.050 for the training size, and F = 5.39, p_{value} = 0.027 for the variability). However, having a systematic curriculum does not seem to lead to better results at the classification or uncertainty test than other teaching curricula (F = 2.78, p_{value} = 0.10 for classification test score and F = 0.13, p_{value} = 0.72 for uncertainty test score).

Findings:
- Participants exhibit various teaching curricula in which the uncertainty measure can be a guide for selecting new training images.
- Participants who adopted a systematic teaching curriculum expressed a better understanding of the classifier behavior. They also provide larger and more variable training sets.

Understanding of differences between aleatoric and epistemic uncertainty

We are now interested in how participants perceive the difference between aleatoric and epistemic uncertainty. The questionnaire suggests a slight difference for the question "The uncertainty measure helped me to identify examples my classifier does not know" in favor of the epistemic uncertainty as shown in Figure 4.18.

Based on the qualitative data, we found that five participants (P5, P6, P10, P13 and P16) claimed that they perceived a difference without being able to express the difference precisely: *«I see that the logic of the A is different from the B but don't know how. The results are a bit different»* (P10). Three participants (P4, P16 and P2) acknowledged that the difference they perceived might be due to a different training strategy of the classifier rather than an intrinsic difference in the way the uncertainties behave: *« In fact, in general I understood uncertainty A less than B, but I can't figure out if that was because of what I recorded or because of*

the uncertainty» (P16). That being said, we found that specific situations triggered notable differences in the way participants perceived epistemic and aleatoric uncertainty. We report these situations and participants' comments in the following subsections.

Figure 4.18. The results from the likert-scale question "The uncertainty measure helped me to identify examples my classifier does not know" suggest that epistemic uncertainty is more helpful to identify novel images than aleatoric uncertainty.

Placing a card in the exact same configuration as a training example would give consistent epistemic uncertainty

Four participants (P3, P8, P9, P16) stated that epistemic uncertainty was extremely low when a card was placed in the exact same position as an other example (from the same class) in the training set. They declared that moving away from this exact position resulted in a quick increase of the epistemic uncertainty. For instance, participant 3 said *«If it's the same place where I took the picture it's completely certain. And when I start to move from the card, the uncertainty rises»* (P3).

This situation can also occur after adding a new image in the training set and leaving this card under the camera. Participant 15 was confused that aleatoric uncertainty was not decreasing significantly when considering the exact same image after the classifier update: *«I don't understand why it's not at 100% certain since I just told it that it's a queen»* (P15-A). These reactions may explain the Likert-scale result presented in Figure 4.18 which suggests that epistemic uncertainty is seen as more useful to identify images that the classifier does not know.

Ambiguous configurations and unstable classification lead to consistent aleatoric estimation.

Most participants explored ambiguous examples by placing two different cards next to each other from different classes. This situation triggers comments regarding the difference between aleatoric and epistemic uncertainty. Since the classifier can only guess a single class, participants commented that the classifier prioritizes one class over another. For example, when participants placed a Nine next to a King (resp. Queen), the classifier usually predicted a Nine and ignored the King (resp. the Queen). This led P2 to wonder about the inner working of the classifier during the aleatoric uncertainty condition: *«I wonder*

if it works with a sufficient minimum, if there is a sufficient minimum of data to say that there is a nine, and it says that there is a nine and so the king here is negligible.» (P2). Still on aleatoric uncertainty, participant 10 said: *« when there is a difficult situation, such as two cards of different types or another new card, for this machine [aleatoric uncertainty condition], it is difficult to be certain, as if this machine is aware of the situation. It can't take responsibility for the answer. The answer is always "I'm not sure"»* (P10).

When exploring ambiguous configurations, participants encountered situations in which the predicted label was unstable i.e. it was quickly changing between two classes despite a stable image in the camera. In this situation, the two types of uncertainty behaved differently. Since the aleatoric uncertainty is based on the softmax predictions i.e. its computation is based on predictions, and the uncertainty level was mainly high in this situation. By contrast, the epistemic uncertainty is computed on the feature space, before the predictions. Consequently, the uncertainty level could be very low in this situation. Three participants expressed their confusion with the epistemic uncertainty, when the classification was unstable. For example, participant 8 said that *«Then it's funny because it switches between queen and king all the time while saying it's certain. It seems strange to me that it's certain about the uncertainty but at the same time the label changes every half second like that»* (P8). In the second iteration with aleatoric uncertainty, participant 8 perceived the difference: *«The first time, it blinked between queen and king and was certain. This time it blinked but was less certain»* (P8).

Image background and participants' hand trigger consistent epistemic uncertainty

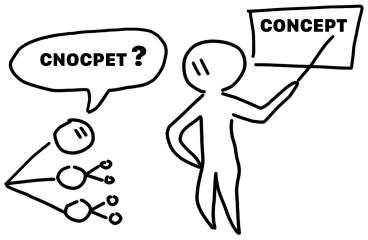
The edge cases of having another object in the image, such as the participant's hand, or having no card at all, also raised comments that differ between the types of uncertainty.

We observed that the aleatoric uncertainty stayed low when a card was presented next to the participants' hand. By contrast, epistemic uncertainty was always high when the participants' hand was next to the card. Five participants (P1, P7, P8, P9 and P10) noticed such behavior in either one or the other condition. Participant 7 placed a nine next to a queen during aleatoric uncertainty condition. When P7 hid the queen with their hand, the aleatoric uncertainty rose: *« And if I put a 9... the uncertainty increases, it predicts that it is a king. If I put my hand on the queen, the uncertainty goes down and it hesitates between a king and a 9. That's a pretty good sign»* (P7). For epistemic uncertainty, participant 1 said: *«For example, when I showed the card with the hand, right away, it gives high uncertainty»* (P1). Participant 9 also

said on epistemic uncertainty that «*It is going to be very uncertain when it's something that doesn't match at all, like the hand. When I tried to put the hand, it was very high because it didn't know at all*» (P9).

In summary:
- The epistemic uncertainty is seen as more helpful than aleatoric uncertainty to identify examples a classifier does not know.
- Differences between the aleatoric and epistemic are perceived in specific situations highlighting the notions of ambiguity and novelty.

4.6 Comparing users' teaching curricula with active learning curricula.



This section discusses our previous empirical results in the light of *Active Learning* (AL). In particular, it presents performance comparisons of a model trained with a simulated teaching curriculum using AL with models our participants trained.

Active learning is a scenario in which the learning model is allowed to be "curious" and can query unlabeled instances on which it will be trained [Settles, 2010]. The human role is contrasted betwee AL and IMT. In AL, the machine chooses training data, and the human delivers annotations, while in IMT, the human chooses training data and the machine delivers its feedback.

At the core of AL strategies lies a measure of epistemic uncertainty. This measure drives the selection criterion: if the uncertainty is too high on a new example, this external information source is queried [Cohn et al., 1996]. AL is designed as a "human-the-loop" method, where a human annotator labels the queries from the algorithm. The model is thus expected to learn faster and with fewer examples by strategically choosing new uncertain examples (according to the epistemic definition of uncertainty).

Active learning (AL) fits most standard tasks in machine learning, such as classification or regression and applies to various machine learning models, from shallow classifiers [Pereira-Santos et al.] to deep neural networks [Gal et al., 2017], as well as a wide variety of real-world problems such as speech recognition [Riccardi and Hakkani-Tür, 2005],

image search [Feng et al., 2012], prediction of protein functions [Xiong et al., 2014] or tomography [Maiora et al., 2014] among others. AL is used in assistive educational systems, in which the ML algorithm model students' knowledge and queries the student with the most uncertain example. This uncertain example is assumed to trigger the most progress from the human student.

AL is generally used to reduce the cost of labeling and achieve good performance with fewer labeled examples. Pereira et al. [Pereira-Santos et al.] and Ramirez-Loaiza et al. [Ramirez-Loaiza et al., 2017] studied AL performances with shallow ML models and showed that AL improves classification performance over standard random learning on average but not systematically. Pereira et al. [Pereira-Santos et al.] conducted a benchmark evaluation of various combinations of classifiers, datasets, and active learning strategies. They showed that performance gains are uneven and application-specific and highlighted the prevalence of the classifier chosen over the AL strategy on the classification performance. Gal et al. [Gal et al., 2017] applied AL to an image recognition task with deep convolutional networks, approximating epistemic uncertainty through stochastic forward passes referred to as Monte Carlo Dropout [Gal and Ghahramani, 2015]. The authors combined AL with an approximated Bayesian deep neural network (using Monte Carlo dropout) and significantly improved the classification accuracy for real-world datasets such as the MNIST dataset [Lecun, Y] and skin cancer diagnosis from lesion images [Codella et al., 2018]. Beluch et al. [Beluch et al., 2018] demonstrated that an ensemble of deep learning models (Deep Ensemble) outperforms the Monte Carlo dropout technique for uncertainty-based AL.

Active learning offers other advantages than performance gains, which is not the main goal of IMT. It allows users to reflect on the model by revealing uncertain regions. However, it is relevant to wonder whether a machine teacher controlling the information given to the system could perform better than passive annotators. Would AL lead to better classification performances than participants' curricula in our interactive machine teaching scenario[2]? The following subsection compares the performance of a model trained with a simulated teaching curriculum using AL and the model our participants trained.

[2] Our ML pipeline includes a pre-trained embedding appended with a trainable multi-layers perceptron (MLP), see Section4.4

Benchmark on active learning teaching curriculum

This section describes the procedure to compute the learning curves of several teaching curricula using active learning (AL). These learning curves use the same models, datasets, and uncertainty estimates

presented in Section 4.3. We analyze the CARDS dataset in order to compare the classification performances calculated with the ones obtained by the participants' models in the machine teaching experiment presented in Section 4.4.

AL spans three data acquisition scenarios, which are summarized in appendix B. The analysis uses the pool-based scenario i.e. the model picks queries among a pool of unlabeled data. This pool comprises the 150 training examples presented in Section 4.3. The model accuracy is computed on the same 20 instances used in the user experiment. We use the MobileNetV1 embedding because it provided the best performance for identifying uncertain images on the CARDS dataset, as demonstrated in Section 4.3. We consider both aleatoric and epistemic uncertainty estimates and a random baseline i.e. a random sampling strategy that picks random examples to be queried.

The seed, i.e. initial training set, comprises three randomly chosen images. The maximal budget of training images is fixed at 30, which is more than what the participants could collect in 7 minutes and 30 seconds. For each sampling technique, 100 different curricula are computed using 100 seeds taken randomly. The test accuracy is calculated for each query.

Results

For each uncertainty sampling strategy, we plot the test accuracy averaged over the 100 curricula according to the budget i.e. the size of the training set. The learning curves are represented on Figure 4.19. Participants' final accuracy and training set sizes are represented with black dots, for each iteration (A or B).

Our results show that most uncertainty sampling strategies perform similarly to the random selection baseline. The performances start to diverge after 20 training instances with around a maximum 5% accuracy gap. More importantly, most participants' classifiers perform significantly better given a budget than models trained with the AL procedure.

- AL performance gains are not significantly higher than random baseline when used with transfer learning, including pre-trained embedding appended with a trainable MLP.
- Participants' teaching curricula exhibit better performance than AL curricula, given a certain budget.

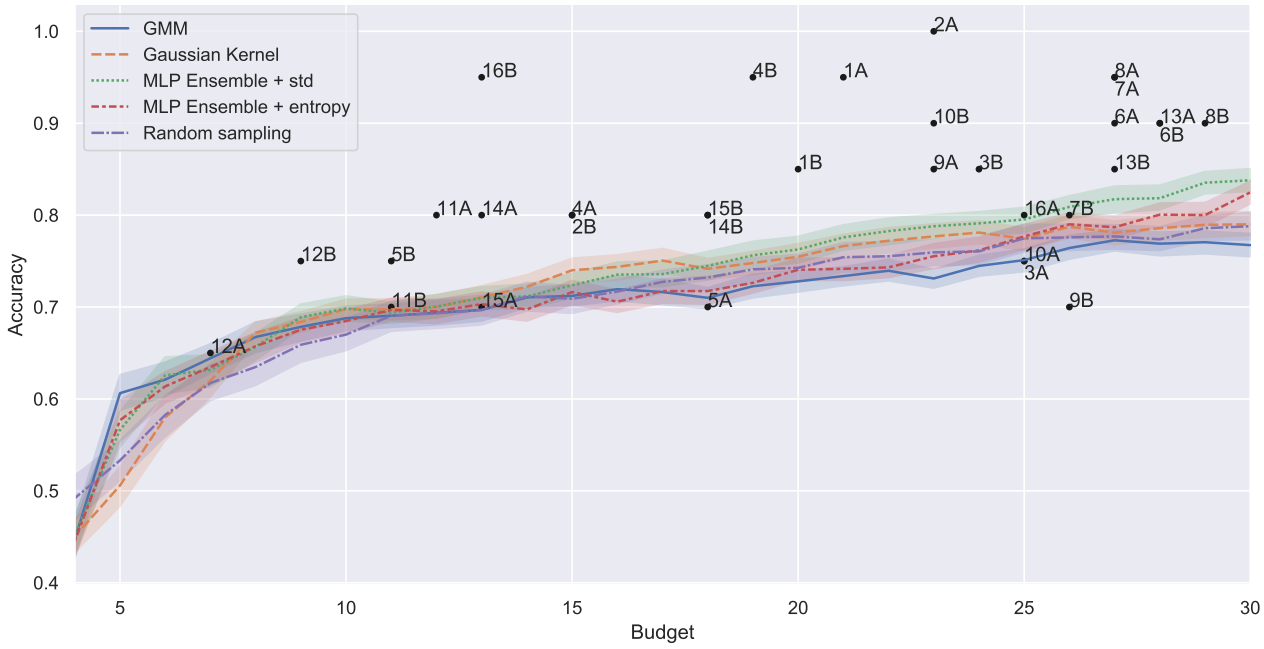


Figure 4.19. Learning curves using pool-based active learning sampling strategies. The participants' model accuracies given their final training set are represented with black dots.

These results are preliminary in that they are follow-ups of the main user study presented in chapter 4.4. An entire study could be devoted to comparing machine-computed and human curricula. However, our precise use case begs an important question: What knowledge a human person provides beyond machine-computed uncertainty that further improves models? In this situation, either our uncertainty estimates are flawed, either we capture human's ability to teach (by organizing curricula with their foresight, insight, and sensemaking) that an untrained artificial neural network alone could not have.

4.7 Summary of the chapter

We explored two types of uncertainty, aleatoric and epistemic, in an interactive machine teaching task with non-expert users. We ran a benchmark study that applied transfer learning techniques to real-time uncertainty estimation. We found that the variability of the data in the feature space is essential for detecting ambiguous and novel images.

We used the results of the benchmark study to design a one-factor, within-participants experiment with non-experts that compares how they use and perceive aleatoric and epistemic uncertainty, both with respect to their teaching strategies and their understanding of the clas-

sifier. We asked participants to teach a classifier to recognize a dozen different playing cards among three classes using an Interactive Machine Learning application. Each participant received real-time classification and uncertainty feedback selected from the benchmark study results. We measured participants' ability to guess how well the classifier will predict new card images, with respect to both classification and uncertainty. We also interviewed participants about their subjective understanding of the uncertainty measures.

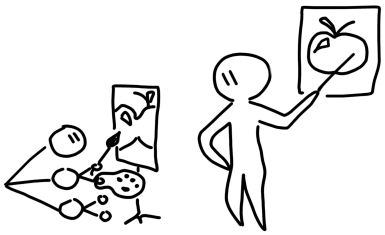
We found that participants' choices made while teaching—especially regarding training set size and variability—are more important than the type of uncertainty participants were exposed to. We also identified and discussed two teaching approaches: the first uses uncertainty to guide the selection of training data; the second systematically introduces variation across the classes. We found that the latter results in a better understanding of the classifier outcome. Finally, we identified three specific situations where participants successfully perceived differences between the two uncertainties, highlighting the notions of ambiguity and novelty in the data.

Lastly, we compared the accuracy of our participants' classifiers with models trained using an active learning procedure. Participants obtained better performances than simulated curricula using AL. Furthermore, AL curricula do not perform noticeably better than a random selection baseline. These results suggest that transfer learning performances improvement prevails over AL ones. Future research should focus on user-centered evaluations of AL in an IMT context, especially mixed-initiative or user-triggered AL.

This chapter brings a human-centered perspective to a theoretical and computational problem—uncertainty estimation in neural networks—that may be beneficial to several fields such as Explainable AI and Interactive Machine Learning.

Chapter 5

Challenges and opportunities for machine teaching in art



This chapter presents two artistic collaborations that resulted in two installations involving machine learning (ML) algorithms. Figures dissidentes in collaboration with the artist Rita Hajj was exhibited in Institut du Monde Arabe during summer 2021. Cor Epiglottae was created in a context of an artistic Hackathon with Hervé de Saint Blanquard, Alexandre Boiron, Elsna Aurand, and Junhang Yu and exhibited in Gallerie Joseph in August 2021. I adopt a reflective perspective and discuss the challenges we face to use ML in those two contrasting art projects. Through these two experiences, I discuss how IMT principles can be challenged when applied to artistic projects.

The research presented in the previous chapters tackles the general public when involved in the teaching image classifiers. This chapter does not report research but personal reflections on artistic collaborations using ML for art installation. For this reason, I adopt a reflective approach to discuss challenges and opportunities to apply IMT in art. This chapter may be of more interest to artists and designers eager to use ML in their work than to scientists.

The usual assessment criteria of ML models are generally inapplicable in art since no objective metrics apply to evaluate the model's outcome. Artists are interested in the generative properties of ML algorithms. Model outcomes are judged subjectively and re-purposed in the narratives of a piece. Indeed, using ML in a piece of art often imply to discuss its use in society and the political connotation associated with ML, such as crowd surveillance with facial recognition [Caramiaux and Donnarumma, 2021] [1].

[1] See for example https://marcodonnarumma.com/series/humane-methods/

Artists are also subject-matter experts, among others. They can have practices that generate artifacts in a systematic way. For instance, artist Ronan Barrot painted a skull on his palette whenever he had a break during his work time. He thus accumulated thousands of paintings of skulls as by-products of his primary artistic production, which led him to collaborate with AI artist Robbie Barrat for the piece *Infinite Skulls* that uses a Generative Adversarial Network (GAN) to generate a unique skull to each visitor[2]. IMT might also empower artists in their activities and be challenged by artists' specific needs, which often rely on new types of tasks (generation), an explorative workflow (i.e. emphasize the importance of making mistakes without consequences), and a subjective assessment of the generated outcomes.

[2] The collaboration and the resulting art piece are described here https://robbiebarrat.github.io/skulls.html

I present two artistic collaborations that resulted in installations involving ML algorithms. I highlight the contrast in their context of creation, collaboration, techniques, and interactions, illustrated in Table 5.1, and discuss my personal insights on how artistic re-purposing of ML could challenge IMT design guidelines.

		Figures Dissidentes	**Cor Epiglottae**
Installation	Type	Visual and sound	Sculpture and sound
	Keywords	Belly dance - Arabic Divas - Bodies - Feminism	Science - Biology - Cybernetic
Collaboration	Context	Spontaneous collaboration	Hackathon
	Number of collaborators	2	5
	Timespan	Spread over a year	Intensive, on 5 days
ML pipeline	Models	Generative DNNs (VAE and RNN)	Real-time regression and classification using stochastic models (HMM)
	Training	Offline using GPUs	In situ, with a Machine Teaching scenario
	Interactivity	None	With the audience

Table 5.1. Contrasts on the type of installation, collaborative process and ML pipeline between the two artistic collaboration *Figures Dissidentes* and *Cor Epiglottae*.

5.1 Figure Dissidentes

Description of the artwork

This section presents the description of the artwork as envisioned and written by Rita Hajj, the leading artist of the project.

Figures Dissidentes is a digital work that investigates the practice of dance known during the golden age of Egyptian cinema under the

Figure 5.1. ***Figures Dissidentes*** **exhibited in** ***Institut du Monde Arabe*** **(IMA) in Paris.**

reductive term of *"oriental dance"*. This work aims to celebrate Arab divas and deconstruct the stereotyped representations they evoke.

This cultural dancing heritage was exposed to the rest of the world through women divas dancers and the movie industry that developed in Egypt in the 1940s. Dancers conveyed freedom far from the social and religious hegemony of the time, in which dancers remained under the control of the movie industry, their husbands, or family. The belly dance showed a distorted reality of an Arab world without sigma on womens' status and freedom.

Figures Dissidentes echoes the aspiration for socio-political changes of the new generations. Dancing is a way to express this joyful and creative resistance. The work reveals the popular, family, and festive anchors of this mode of expression. Fed by movie scenes and amateur practices, *Figures Dissidentes* plays the role of a mimetism of the Arab

cultural memory. The installation shows a video generated by an artificial neural network trained on a corpus of both dance scenes from the 1940s movies of the golden era and amateur dancing videos taken from social networks such as TikTok. The first DNN is a Variational Auto-Encoder which is trained to deconstruct and reconstruct each video frame. The second DNN is a recurrent neural network (RNN) that is trained to learn the temporal dependencies between these images and the motion. The juxtaposition of the raw videos and noises incorporated in the neural network introduces transitions that blur the lines between the past and the present, between bodies of different eras and genres.

Context of creation

Rita Hajj received a Friends Prize from the *Institut du Monde Arabe* (IMA) in Paris, from which she obtained a grant and three months of an artistic residency in the *Cité internationale des Arts* in Paris. Rita Hajj initiated the collaboration, who was looking for technical assistance on ML algorithms for generating videos. The collaboration was spread over a year and a half, from January 2020 to May 2021. The covid-19 pandemic extended her stays in Paris and postponed the inauguration of the piece to May 2021, during an exhibition dedicated to Arab divas[3].

[3] https://www.imarabe.org/fr/expositions/divas-arabes

Collaboration specificities

Soon after starting the project, our collaboration became fully remote due to the pandemic. My work on the project was conducted during my free time in parallel with other research and teaching activities.

The pipeline we set up for exchanging data inputs and model outcomes is illustrated in Figure 5.2. Rita Hajj and I shared a common repository hosted by the DropBox company. Rita Hajj was uploading new samples from her corpus i.e. movie scenes from the 1940s golden age of Egyptian cinema, and amateur videos are mainly taken from the TikTok social network. The neural network computations were made on a computer equipped with Graphical Process Unit (GPUs) located at the university. I uploaded the new video samples once new generations were performed, and we discussed the results through various messaging applications, either by phone, skype, or email.

The development of *Figures Dissidentes* suffered from a heavy data pipeline in which Rita Hajj curated the model inputs but could not access the model or its outcomes directly. Moreover, the main artist

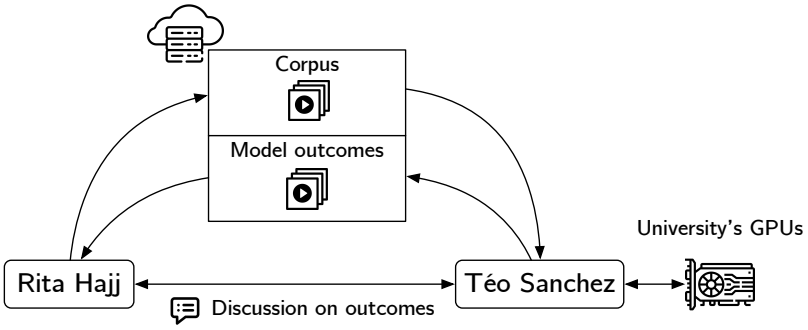


Figure 5.2. Schema of our remote collaborative process during the pandemic.

could not directly change the model parameters nor trigger a new model training. In our collaboration, I sometimes played the role of a proxy between the artist's intents and the model. I tried to interpret subjective feedback on the visual aesthetics of the outcomes and turn them into actionable modifications of the model parameters. I believe these obstacles are widespread in art-science collaborative works. I believe these issues could be partially solved with an adequate framework. Even though Marcelle is only applied to classification yet, it provides the building blocks to improve collaboration between artists and ML practitioners: the possibility to create customized interfaces and interactions that share data and models on a common object (the datastores). Developing the necessary interactive applications prior to the collaboration (even using Marcelle) would have taken a crucial amount of time away from the main project.

Machine Learning pipeline

Rita Hajj wanted to preserve recognizable motions of the dancers as well as using the corpus video as raw material. These prerequisites constrained the choice of the model since usual trajectories in GANs' latent space might not provide a sense of dance movements. Far from being an expert in generative models, I conducted literature review to explore generative models that could learn temporal dependencies (the movement) on raw videos and that could result in the highest resolution possible given the computational resources at my disposal. We agreed on an ML pipeline that involves videos as inputs and outputs and implies two different DNNs, one to learn how to reconstruct the video frames (spatial reconstruction) and one to learn how these video frames sequence in time (temporal reconstruction).

- The first model is a Variational Auto-Encoder (VAE). It jointly learns to encode each video frame (no matter in which order) in a latent

space[4] and decodes the latent vector into a reconstructed image. VAE belongs to the family of probabilistic graphical models, which are graphs (the neural network) that learn the conditional dependence structure between random variables, the original image, and its reconstruction.

[4] The latent space encodes each original image into a vector of lower dimensionality

- The second model is a Recurrent Neural Network (RNN) using Long Term Short Term (LSTM) memory units. It is intended to learn the temporal sequence of the video frames in the latent space, which is a simpler task than learning temporal dependencies in the original pixel space. It takes the latent vector of each video frame sequenced in time and tries to learn these temporal dependencies.

This model architecture was proposed and experimented with by the research scientist Arthur Juliani [5]. The model architecture chosen is illustrated in Figure 5.4. The VAE and LSTM are trained independently in this pipeline. Earlier model architectures replace the encoder and decoder layers (grey) with LSTM layers (green) to train the image reconstruction and temporal dependencies together. After training, new video samples can be generated using the trained RNN and the VAE decoder, as illustrated in Figure 5.4.

[5] https://github.com/awjuliani/NeuralDreamVideos

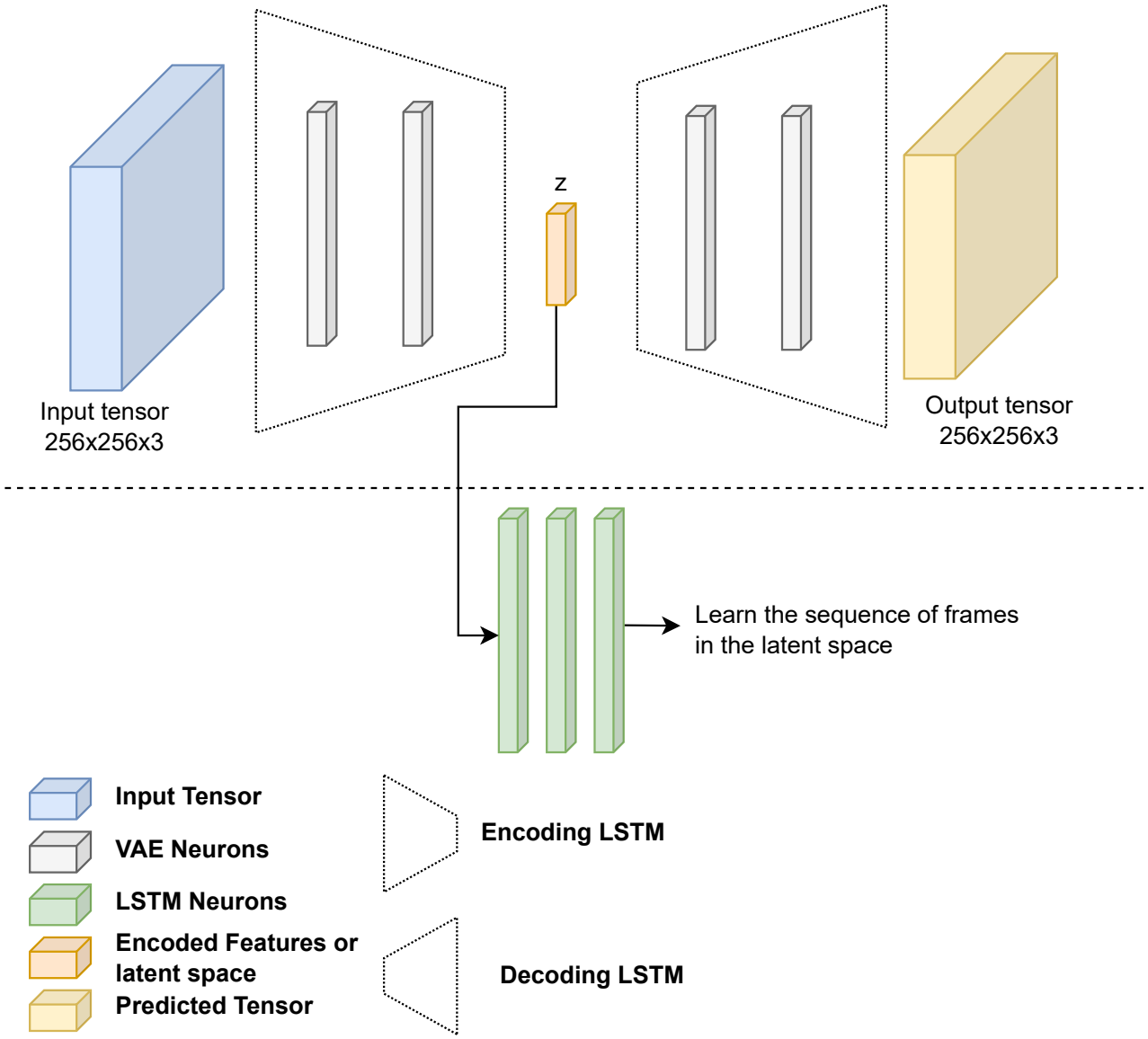


Figure 5.3. ML model training pipeline uses both a Variational Auto-Encoder (top) and a Recurrent Neural network (bottom). Both models are trained independently.

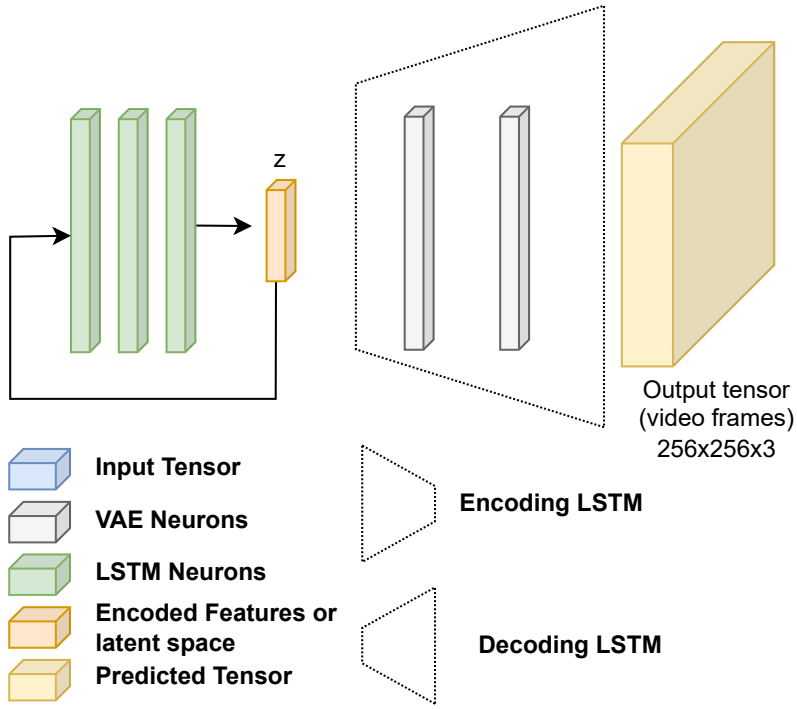


Figure 5.4. ML model inference pipeline only use the Recurrent Neural Network and the Variational Auto-Encoder decoder to generate new video frames.

We decided to stick to this architecture and tried to exploit it by adjusting hyper-parameters and iterating on the resulting generations. Generative ML models exhibit many hyper-parameters that might affect the aesthetic of the results. The hyper-parameters of our configuration are listed in Table 5.2.

We explored the generative space trying different hyper-parameters by trial and errors. While the hyper-parameters of the VAE affected the aesthetic of the static image, the hyper-parameters of the RNN affected the motion across different images. The material generated was edited and assembled by Rita Hajj for final rendition. Figure 5.5 depicts intermediate results along this trial-and-error process.

Figure 5.5. Snapshots of intermediate results along the trial-and-error process. The image reconstructed could be to too sharp or too blurred, and the learned movement (succession of frames) could change greatly depending on the choice of the RNN hyper-parameters.

	Variational Auto-Encoder	Recurrent Neural Network
Architecture	Size of the latent space	
		Type of memory units
	Number of hidden layers	Number of hidden layers
	Number of neurons per layers	Number of neurons per layers
Optimization		Length of the input sequence
		Decay rate
	Learning rate	Learning rate
	Batch size	Batch size
	Epochs	Epochs

Table 5.2. Model hyperparameters that could possibly be adjusted in the architecture chosen.

Subjective insights on the collaboration and prospects for IMT in art.

Frustrations resulted from the use of ML models in this artistic collaboration. The trial-and-error process was frustrating in the ML pipeline because the aesthetic changes when trying new training configurations were neither predictable nor sufficiently large in magnitude. The aesthetic we obtained with a VAE was restricted to a narrow domain. I later understood that VAE is **mode-covering** while GANs are **mode-seeking**, which explains why they result in such different aesthetics. This distinction that constrains the resulting visual aesthetic of VAEs and GANs is discussed in Appendix C. One may wonder why a VAE model was chosen over a Generative Adversarial Network (GAN), which is a popular generative model for art and design [Goodfellow et al., 2016]. I was discouraged from trying another generative architecture because I had already invested too much time and effort in exploiting the VAE-RNN architecture, even though its results were not entirely satisfactory regarding motions and resolutions. This fear of making a step backward is a well-known cognitive bias called *sunk-cost fallacy* or *escalation of commitment*. I believe the computational and memory cost of generative deep learning architectures might affect end-users will to explore model alternatives. Second, GANs are difficult to train because they rely on two deep neural networks trained simultaneously. A brief description on the training process of GANs is also given in Appendix C. Third, the training procedure of GANs does not include an image encoder that can encode the video frames in lower-dimensional space. Consequently, learning time dependencies between the video frames is challenging with a GAN. I later discovered the existence of a VAE-GAN architecture that both encode an image in a latent space and decode the image using a decoder trained with the adversarial pipeline [Larsen et al., 2016].

On the collaborative aspects, the frustration resulted in the lack of agency from the main artist, Rita Hajj, on the training process and resulting models. It is striking that applying IMT concepts to generative models is challenging for several reasons.

First, the heavy architectures of generative networks do not afford fast iteration cycles on the model training, which makes difficult to apply the IMT principles. As seen in previous chapters with classification problems, we could expect transfer learning to leverage these issues. However, transfer learning and pre-trained models raise essential questions in the creative process. In a thread on Twitter, the artist Alisson Parish highlighted the problem of pre-trained models regarding authorship. To her, understanding a language model's training set is essential for understanding its predictions. Furthermore, she argues that using publicly available pre-trained models prevents her from accessing and understanding the "voices" with which the model is speaking. She emphasizes that large models do not generate but reflect other people's voices. Understanding and owning her training set allow her to judge if she is morally and legally authorized to speak with those voices. Alisson Parish's point might particularly applies to generative models for natural language. Ownership problems also lie in the available infrastructures and research directions taken which do not always promote lightweight architectures[6]. Artists need easy-to-compute architectures that preserve a high degree of personalization on idiosyncratic data. Transfer Learning using publicly-available pre-trained models does not seem to offer a way out since it conflicts with the need for authorship and customization of the generated outcomes and suggests new research directions for HCI and ML research.

[6] In a tweet from 2019, the artist Helena Sarin argues that «*it feels less and less likely that there will be any research around deep learning on small compute/less data - Google, Facebook et al. are not interested, they want everybody to use their cloud and pre-trained models.*»

Second, very little research focuses on the interpretability and explicability of generative models, which is central to designing meaningful teacher-student interactions according to the IMT framework. In our case, it would have been helpful to be guided to understand how the hyper-parameters affect the resulting visual aesthetic.

Finally, the termination criteria, which deal with the aesthetics of the generated images, can not be expressed clearly as in classification or regression tasks.

On this last point, it is well-known that generative models (especially large GAN architectures) can lead to interesting outcomes at the intermediate training state. Hyper-parameters cannot be tuned automatically to find this optimal training state. Artists then develop know-how by trials and error and look for parameters on which they sub-

jectively obtain interesting and novel aesthetics[7]. Researchers have a role to play in developing and maintaining this artistic and algorithmic cultural heritage. First, HCI and AI researchers could provide more interactive, composable, and appropriable generative models for artists. Quickly browsing and benchmarking several generative architectures could have been beneficial to our collaboration. Second, original artists' tweaks and workflows with ML models should be documented because it is a new form of artistic heritage and can inspire the design of new creative interactions for controlling generative models.

I believe that these collaboration challenges described in this section should be addressed outside of an art-science project but as an HCI problem that would try to apply IMT design principles to generative art.

[7] For instance, the artist Vadim Epstein published several state-of-the-art generative models on his GitHub https://github.com/eps696/stargan2, along with personal findings for tuning hyperparameters. On intermediate model states, he stated:
«Model weights may seriously oscillate during training, especially for small batches (typical for Cycle- or Star- GANs), so it's better to save models frequently (there may be jewels). The best-selected models can be mixed together [...] for better stability.»

5.2 Cor Epiglottae

Description of the artwork

Figure 5.6. Visitors communicating with *Cor Epiglottae*, during the first exhibition in Gallerie Joseph, Paris.

Inspired by cybernetics and ASMR (Autonomous Sensory Meridian Response), *Cor Epiglottae* is an interactive sculpture mimicking a living organism that reacts to the audience's sound stimuli. The visitor can make sounds to communicate with the sculpture, which will re-

spond by purring, moaning, or screaming. The visitor must come close enough to the sculpture to engage in this dialogue that leaves freedom for interpretation.

The installation learns the stimulus-reaction association using a machine learning algorithm that maps the recorded sound qualities (voiced vowels, sibilant consonants, whistling consonants, etc.) to various reactions. The sound emitted by the sculpture uses a real-time physical model of vocal folds, glottis, and mouth called Cantor Digitalis [8] [Feugère et al., 2017].

[8] https://cantordigitalis.limsi.fr/

Context of creation

This work was created in collaboration with Elna Aurand (design), Alexandre Boiron (art), Hervé de Saint Blancard (art), and Junhang Yu (HCI and design) during the first edition of an artistic hackathon entitled the creartathon[9], organized by the Université Paris-Saclay, Inria Saclay and Societies. This hackathon gathered seven teams of five students from art, design, human-machine interaction, or machine learning who spent a week brainstorming, designing, and implementing an interactive and intelligent object. Each team was advised by professional artists, Fablab managers, and HCI and AI researchers. *Cor Epiglottae* is the result of this short but intensive collaboration, articulated around the notions of intelligence and interaction.

The creation of the work involved several phases. The first days were dedicated to brainstorming. The existing artistic practices of the artists in the team, Alexandre Boiron[10] and Hervé de Saint Blanquard[11], strongly influenced the design space. The ideation process included various sketches, models, and generated images from large text-to-image ML models depicted in Figure 5.8. The ideation ended with the realization of a short video prototype explaining the envisioned artwork. The video prototype can be seen at https://vimeo.com/591541265/00a484989c.

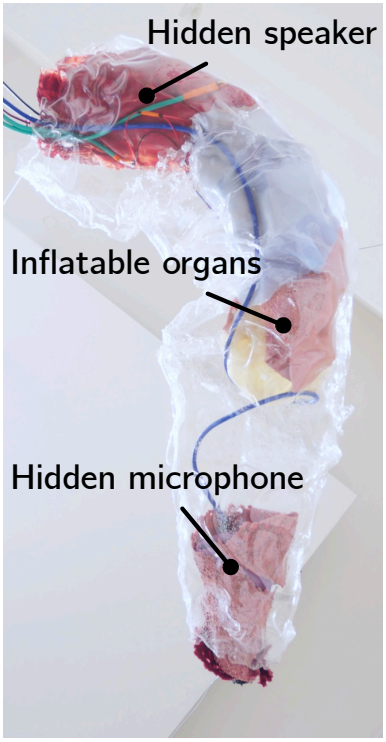


Figure 5.7: Dissection of *Cor Epiglottae*, seen from below.

[9] https://creartathon.com/

[10] https://www.instagram.com/alexandreboiron/?hl=en

[11] https://blancardsuperstar.com/

Figure 5.8. Artifacts resulted from the ideation process. (Left) The first sketches, drawn by Junhang Yu, (Center) text-to-image generation using a VQGAN model, (Right) mock-up of the spatial disposition of the artwork, realized by Junhang Yu.

The roles were naturally divided according to each person's expertise during the design phase. Elna Aurand, Alexandre Boiron, and Hervé de Saint Blancard were in charge of the shell made with a melted acrylic sheet, the organs made with sewn silicone tissues. Alexandre Boiron and Junhang Yu built the pumping system to inflate and deflate the organs to make the installation more alive. I was in charge of the software development i.e. the sound processing, real-time ML regression using the XMM library in MaxMSP [Françoise et al., 2014], and the association to the physical model for voice synthesis [Feugère et al., 2017].

Figure 5.9. Organs artefacts obtained by molding silicone on different objects and surfaces.

Sound processing and machine learning pipeline

Software development was steered by the necessity to have real-time responses to the sound stimuli from the audience. For this reason, the installation embeds an algorithm developed with Cycling'74 Max, that maps Mel-frequency cepstral coefficients (MFCCs) from the recorded audio to the parameters of the Cantor Digitalis, a performative singing synthesizer using a physical model of the vocal folds, glottis, and mouth. Cantor Digitalis was developed between Sorbonne Université and Université Paris-Saclay [Feugère et al., 2017].

This mapping is learned using the XMM [Françoise et al., 2014] library included in the MuBu toolbox [Schnell et al.]. Jules Françoise developed XMM for movement interaction in creative applications. It implements an interactive machine learning workflow with fast training and continuous real-time inference. It includes various models, such as Gaussian Mixture Models and Hidden Markov Models, usable in Cycling'74 Max.

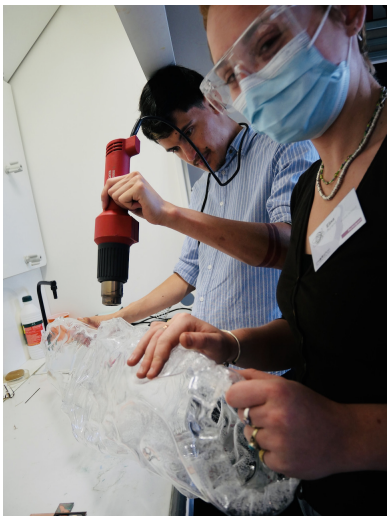


Figure 5.10: Manufacturing of the shell of *Cor Epigloatte*, with acrylic melted with a heat gun.

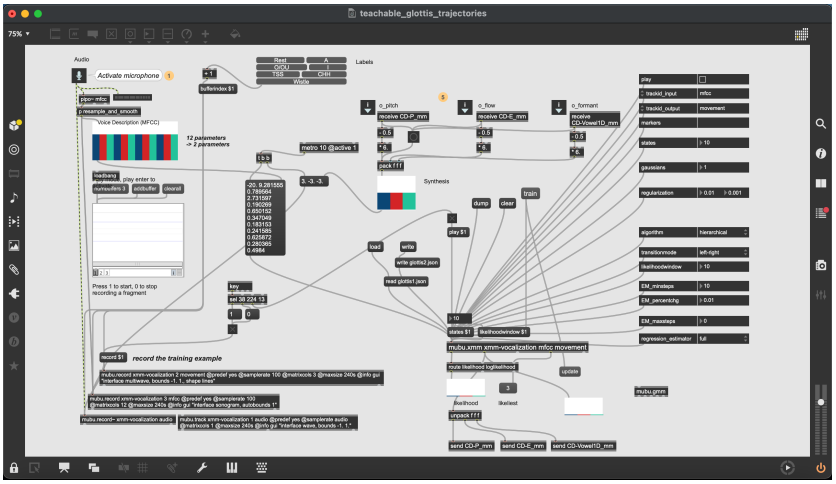


Figure 5.11. Visual programming patch (Max) made for the project using the XMM library [Françoise et al., 2014].

The first version of the algorithm used Hierarchical Multimodal Hidden Markov Models for continuous mapping. The algorithm learned by simultaneously taking sound examples from the audience along with synthesizer trajectories performed on the Cantor Digitalis interface depicted in Figure 5.12, that controls the pitch and vocal intensity of the simulated voice. Performing both the stimuli from the stimulus and the reaction was cognitively demanding, so we curated training examples in duo, as illustrated in Figure 5.13. Hervé de Saint Blan-

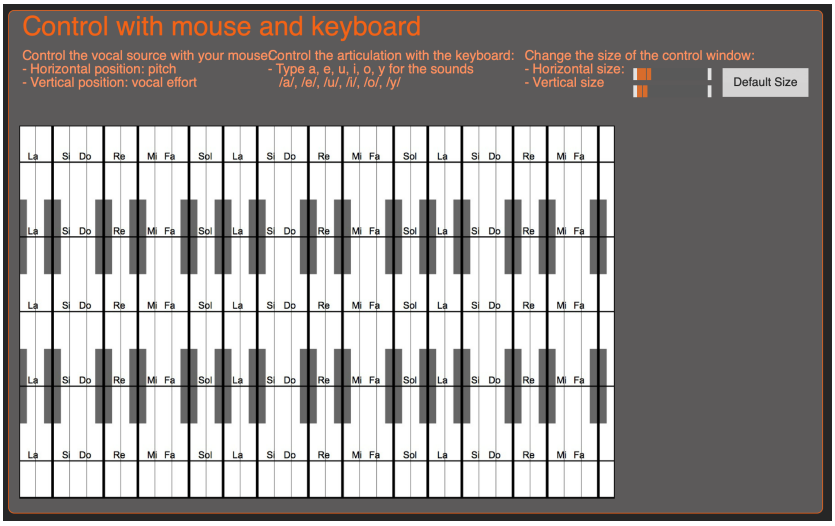


Figure 5.12. The Cantor Digitalis can be controlled with a graphic tablet or a mouse and keyboard. The musician can navigate this two-dimensional space where the x-axis corresponds to the pitch and the y-axis to the vocal intensity.

quard provided sounds while I was simultaneously performing with the Cantor Digitalis.

In summary, the recorded audio is first transformed into MFCCs, which are mapped to synthesis parameters with the ML model from the XMM library. Finally, the synthesizer outputs are sent back to a speaker hidden in the organs of the sculpture.

Interactive Machine Teaching in Cor Epiglottae

It is worth mentioning that the training approach involving two persons falls within the scope of Machine Teaching. Users (the installation designers) can curate their own data and incrementally evaluate the learning progression of the system. However, it moves away from typical applications seen before since no annotations were involved. Instead, the teaching resembles a duet performance, which is illustrated in Figure 5.13. The teacher-learning relationship is similar to imitation learning in that the system learns a behavior policy from demonstrations.

In the vein of the cybernetics, *Cor Epiglottae* is sensing its environment and reacting to it in real-time, which sometimes resulted in a feedback loop. Indeed, since the microphone and the speakers are not too far apart, the system sensed its own sound emissions. We did not consider that in the first attempt to teach the system. The system stayed locked in overreacting states, similar to a Larsen effect. Thus, we had to curate examples of its own sound for the system to have a stable behavior. It is worth noting that we had to retrain (finetune or from scratch) the

Figure 5.13. The teaching process of *Cor Epiglottae* involved a synchronized interpretation of both the input sounds performed by Hervé de Saint Blancard (left) and the output sounds played on the Cantor Digitalis by Téo Sanchez (right).

system each time the installation was moved because the training set was not suited to the new acoustic properties of the room, leading to unstable reactions from the system.

This work leads to challenging considerations regarding IMT, both in terms of the teaching scenario and the system limitations to adapt to new situations. The installation required two persons to train the model synchronously and training adjustments after "deploying" the installation in new acoustic environments. It is worth noting that the system was not learning along with the interaction with the audience durig the exhibition. Improvements could focus on designing more adaptive behaviors that can learn to adapt to new acoustic environments or stimuli over time. Many artists tackled this concept of evolution and learning with installations that learn from interactions with various entities during the exhibition. For instance, Aglaopheme was a robotic guitarist invented by Nicolas Anatol Baginsky in 1992. The robot learned to play by listening to its environment and its own play for several years. The installation *B-612* from the polish artist Natalia Balska is a reinforcement learning algorithm that learns to optimally share water with a living plant and get rewards or penalties according to the health indicators of the plant. These artistic installations adapt to a situation initially staged by the artist throughout the exhibition. Of course, the system's progress should not be too slow, too fast, or too easily predictable to be captivating. Artists investigating learning installations certainly test and tune the learning behavior of their creations in advance. I believe this situation opens exciting prospects regarding IMT because artists do not only want to convey information to the system but also perform a "meta-teaching" by defining how the system will learn over time.

5.3 Summary

This chapter takes a reflective and discusses the challenges and opportunities for applying interactive machine teaching applied to art.

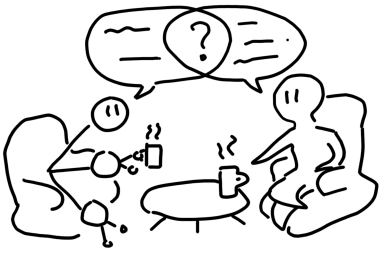
«Figures Dissidentes» addresses the problem of learning movement from a corpus of raw video, aiming at a specific aesthetic that focuses on movements rather than bodies. This is a challenging problem for an ML practitioner since the objective cannot be clearly expressed, and such models imply heavy architectures. We saw that the choice of a generative model strongly conditioned the aesthetic of the results. I believe IML research should provide tools to explore different generative model architectures more easily. Artists and ML researchers would also benefit from documenting artists' innovative tricks to tune generative models to produce novel aesthetics.

The second project, *Cor Epiglottae* is intrinsically different since it does not involve neural networks but lightweight probabilistic models that run in real-time. The installation is also interactive and requires to be "taught" before each exhibition. The system challenges offer interesting prospects to design IMT systems for artists to control how an ML-based artistic installation will learn from its environment throughout an exhibition.

My collaborators and I were in charge of training ML models for artistic installations with atypical termination criteria. This specificity might be less common in other fields of expertise (e.g. medicine, law, science etc.). The challenges we face to convey aesthetic concepts or specific behaviors to the machine highlight promising directions for applying IMT concepts in art and designing collaborative tools for art-science projects involving ML.

Chapter 6

Discussion



This chapter discusses our findings on people's understanding and teaching strategies of image-based ML classifiers in an IMT scenario. In particular, it discusses the teaching strategies elicited by our participants as well as the role of uncertainty in IMT. I finally discuss the socio-cultural implications of this research, such as the use of IMT to leverage peoples' literacy about ML.

6.1 Consolidation and exploration in interactive machine teaching.

In the two user studies presented in chapters 3.4 and 4.4, our findings mostly stem from studying a particular case of teaching: the training of image classifiers in which data are dynamically generated by human teachers. Therefore, model evaluation arises from the ability to create data and get immediate predictions about them. This process blurs the frontier between model testing and training. The quality of the model is assessed without any performance metrics calculated on an existing test set. Participants' verbalization suggests that the creation of an image can carry different users' intents.

An image can be created **to consolidate the fundamentals of the concepts** i.e. participants create examples that the model should know and be confident about. If the classifier fails on these examples, the example is generally added to the training set.

An image can also be created **to explore the boundaries of concepts**. The boundaries of the concepts can be explored in two directions. The more frequent is to **challenge** the concepts with examples that investigate novel areas that carry epistemic uncertainty. The less frequent

exploration is to **precise** the boundaries between two or several concepts, which corresponds to areas carrying aleatoric uncertainty i.e. ambiguity regarding the concepts.

These intents echo the teaching phases elicited by experienced machine teachers in Wall et al. [Wall et al., 2019] and Ramos et al. [Ramos et al., 2020]. Consolidation corresponds to the *cold start* that roughly defines the representative images of each concept (also referred as test-driven machine teaching [Yang et al., 2018b]), while the exploration includes the *challenge* and *boundaries* phases. Our scenario differs in that the *testing* phase is not a separated phase but is incorporated all along the teaching process due to the real-time predictions and uncertainty feedback from the system. As the curriculum develops and the classifier improves, images that were used to explore might become those used to strengthen the model. For instance, in the study from chapter 3, we found that participants that investigated geometrical operations also taught the system with the transformations created. In other words, the tight coupling between exploration and consolidation allows participants to use variability as a way to both 1) challenge the algorithm with ambiguous or novel examples and 2) extend the generality of the taught concepts.

Hong et al. [Hong et al., 2020] also documented the use of variability. However, their task involves separate training and testing phases, which encourage fixed training strategies (also highlighted in [Oh et al., 2020]). These two studies do not involve users as teachers since they have no direct way to inspect the model progress when providing examples. Surprisingly, the authors noticed that testing examples were less variable than training examples. We can suppose that direct feedback and incremental training influences data variability by arousing participants' curiosity to challenge the taught concepts.

It is not clear how the trade-off between *consolidation* and *exploration* improves human teachers' functional mental model of the learner. Our results suggest that participants' explorations can trigger either relevant or erroneous insights about the model learning behavior. I suspect that participants who explore the boundaries of the concepts without consolidating fundamental examples might have a worse mental model than people with a curriculum that gradually incorporates complexity and variability in the training set. Relevant insights might arise when participants explore after ensuring a solid understanding of the models' capabilities on basic examples.

Any new insight about the model, either relevant or erroneous, is

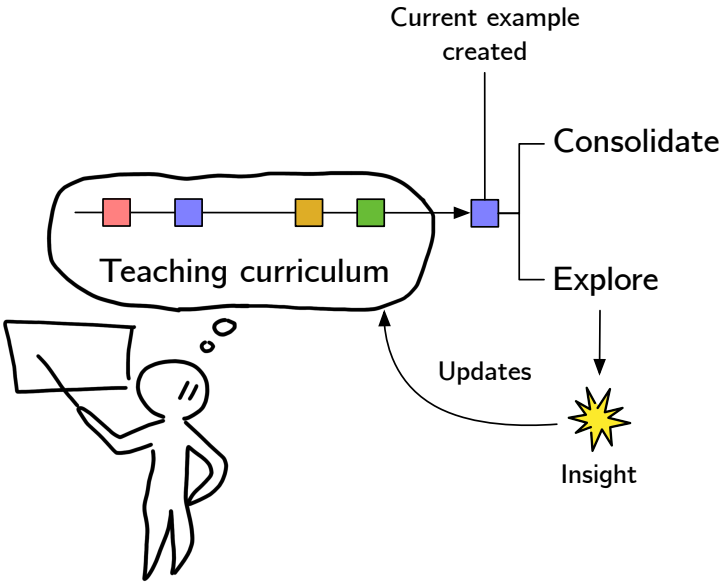
likely to drive participants to update or refine their **teaching curriculum** and **decision-making rules** (i.e. participants' criteria to discard or add a created image to the training set). For instance, it is common among participants that consecutive incorrect predictions on examples from a concept may lead the machine teacher to conclude that the model is not trained enough in this class. Consequently, the teacher might update its curriculum to include more examples of that concept. Alternatively, if complex examples (e.g. used to explore the models' boundaries) are correctly predicted, the machine teacher could think that the model is actually more robust to variability than expected. The teacher may subsequently decide to allow for more difficult examples to be added to the training set. This situation was central to participant 12's strategy reported in section 3.5. However, this situation might occur more frequently in an advanced phase of the teaching, when the model is already robust on variable examples. Figure 6.1 illustrates these two types of updates (curriculum and decision-making rules) described here. The top figure shows how new insight obtained by exploring model blind spots can affect the curriculum envisioned by machine teachers. Note that a participant may not have an well-established teaching strategy in mind. The bottom figure represents how insights can influence participants' decision-making rules i.e. their criteria to decide if the image created should be added to the training set. Indeed, an image belonging to the concept may not be added if the participant judges that it could fool the algorithm [1]

[1] For instance, in the study 3.4, participant 8 said: «*I don't know if I'm confusing him by trying to make things too specific or not. Do I keep it simple so he can understand something simple or do I push him a little bit and try to get him to differentiate things a little bit harder?*»

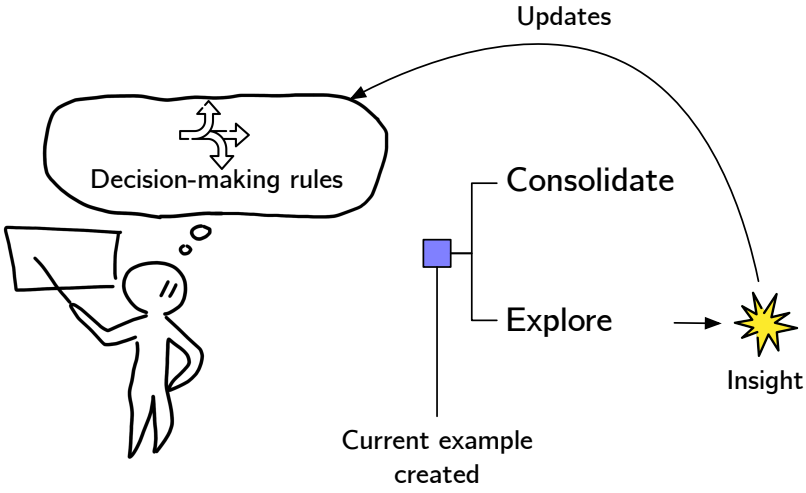
We assume that different data acquisition scenarios and interaction techniques might lead to variation in the teaching strategies regarding curriculum planning and evaluation. For instance, participants can have access to existing datasets of images or documents or provide a batch of data to be learned at once (as implemented in Teachable Machine [Carney et al., 2020a]). Such scenarios provide a different way to assess the model and correct its behavior. As a research field, LMI needs to articulate general and application-specific principles of how people should convey new knowledge to a learning system. This thesis contributes to that effort for scenarios involving sequential learning and user-curated data.

6.2 On the use of uncertainty in interactive machine teaching

Participants using uncertainty as a guide did not train more accurate models, nor were they more able to correctly foresee their classifier



(a) Teaching curriculum updates



(b) Decision-making rules updates

Figure 6.1. Observed behavior on how people update their curriculum and decision-making rules.

outcomes. Instead, participants with a systematic way of curating the data usually provided more and more variable data. Consequently, they trained a more accurate classifier and better predicted their classifier's outcomes regarding both classification and uncertainty estimation. To explain this result, we hypothesize that participants opting for a structured curriculum might have a better mental picture of the content of their training set. Hence, encouraging structured rather than uncertainty-based curricula might be more indicated if we want to improve users' general understanding of their classifier.

Should we then consider uncertainty as unimportant and dismiss its presentation during a teaching session? Our suggest that the two-level distinction in ML uncertainty might not be necessary when the system is trained from scratch to reach a reasonable accuracy. However, I would argue that further research is needed to fully understand the role of uncertainty in IMT. First, uncertainty should be evaluated in other teaching scenarios than sequential learning. For instance, displaying uncertainty evolution over a batch of data might be a powerful tool for users to understand ML models. Second, uncertainty might be helpful in refining the model on more complex tasks involving a larger number of classes and a model evolution on a longer time scale.

The utility of uncertainty estimates might change if users are (1) consolidating the fundamental examples of the concepts by training examples on model blind spots that are easy to find or (2) exploring the tail of the uncertainty distribution (extreme values) to explore the boundaries of the model. On the one hand, filling the model's blind spots is easy but time-consuming. Indeed, our results in study 3.4 show that several participants exploited geometrical operations (rotations, translations, changes in size) to investigate models' invariance and augment the training set. On the other hand, accessing the tail of the uncertainty distribution is difficult in teaching scenarios that do not involve large unlabeled data a priori. For both situations, we foresee promising research directions in integrating data augmentation guided by epistemic uncertainty in order to generate these edge-cases examples from previously given examples. Such a process could make machine teachers more efficient in consolidating the concepts and exploring the tail of the uncertainty distribution.

6.3 On the use of active learning in interactive machine teaching.

This section discusses the results presented in Figure 4.19 in the light of the existing literature that applied an active learning scenario in a teaching task.

As presented in related work, the teaching task considered in the HRI articles when assessing active learning is very different from ours. In [Cakmak et al., 2010], the task only has 552 possible inputs, and a concept comprises between 10 and 28 possible positive examples. The input space is limited because examples are composed of discrete characteristics (shape, size, and color). In our case, our classification task has an infinite input space, and a concept can comprise an infinity of examples. Our ML pipeline involves transfer learning techniques to make deep learning models quickly adaptable to users' inputs. In this situation, we observed that simulated performance gains with AL are much less significant and systematic than with a discrete input space, which corroborates the observations from Pereira-Santos et al. [Pereira-Santos et al.].

Without transfer learning, deep learning accuracy gains using AL are significant but are only about 2% (MNIST) or 5% (CIFAR) considering budgets between 1000 (MNIST) or 10000 (CIFAR) [Gal et al., 2017, Beluch et al., 2018]. In addition, the entire ML model is retrained from scratch at each query, which prevents rapid iteration on the model training. The performance gains, training costs, and budgets envisioned in these traditional ML publications are far from applicable in an IMT scenario and might not even be perceived by participants.

Furthermore, AL requires a large pool of unlabeled data, which users do not have when creating the data from scratch, as in our IMT scenario. This constraint also applies to stream-based AL, whose effectiveness relies on a low-cost acquisition of stream data. These reasons confirm that AL might not always be applicable with data acquisition scenarios encountered in interactive machine teaching and should not be used for performance gains in IMT.

Detailed user-centered evaluations of AL in IMT remain to be done. The query labeling process can disrupt the incremental workflow of IML and make users lose control of the teaching process. Hence, mixed-initiative AL or user-triggered AL (as envisioned in Ramos et al. [Ramos et al., 2020] in which AL is used as a data sampler) are promis-

ing interactions that might improve users' teaching curricula and encourage users to reflect on the systems' knowledge and gaps.

6.4 On the use of deep learning in interactive machine teaching

The use of deep learning (DL) models is criticized in human-centered AI and in IMT for their lack of intelligibility [Ramos et al., 2020, Rudin, 2019], especially when involved in high-stake decision-making applications (e.g. in medicine or law). DL models are opaque by design because the learned knowledge is dispersed among the neurons' weights. Furthermore, our results suggest that participants were confused about neural networks properties, as reported in 3.5. Designing actionable explanations with DL is possible [Schramowski et al., 2020] but requires other models, complicating the overall pipeline.

On the opposite, inherently intelligible such as linear models, rule-based algorithms and decision trees are preferred because they offer actionable parameters that humans can understand and verify. IMT suggests enabling users to decompose semantic features into sub-features and compose inherently intelligible models into a schema. This modular approach would decompose problems into simpler and more transparent ones. I would argue that DL models can fit in this modular vision developed in IMT [Simard et al., 2017, Ramos et al., 2020].

Training a DL model can take several hours, which is a main problem for human-centered AI and a stalemate for IMT. However, the active field of transfer learning (TL) [Niu et al., 2021, Weiss et al., 2016, Pan and Yang, Tan et al.], which is briefly introduced in appendix A, can enable the use and expressive DL with rapid training. TL relies on reusing trained networks (e.g. the first layers of neurons) to speed up the training on a new task. Mishra et al. [Mishra and Rzeszotarski, 2021] showed that transfer learning concepts are accessible to non-experts with appropriate interactive tools. They designed a prototype in which users can stack pre-trained neural networks to perform image classification with transfer learning. As in this thesis, participants develop strategies, sometimes ineffective, to perform the TL task. The authors point out that inaccurate perceptions of the system's progress can impede their ability to perform the task. These results show that, as in electronics, ML models could not only be composed in parallel (schema) but also in series (transfer learning) to personalize the behavior of a system on a new task. This serial composition illustrated by

Mishra et al. [Mishra and Rzeszotarski, 2021] is particularly effective and developed with artificial neural networks.

The modularity of AI systems echoes the rivalry between the connectionist and symbolic approaches of AI in the 1960s. The *symbolic* approach considers that the mind does not directly access the world but acts on intermediate representations (i.e. semantic features and schemas). These representations must be described with symbols that algorithms can manipulate. On the other hand, the connectionists consider that the mind is an emergent property of the interconnection between neurons (i.e. artificial neural networks and deep learning). The duality between the *connectionism* and the *symbolism* might often be exaggerated, and symbolic AI researchers do acknowledge the contribution of the connectionists such as Frank Rosenblatt. In the preface from the 1988 edition of *Perceptrons: an introduction to computational geometry* [Minsky and Papert, 1969], Seymour Papert and Marvin Minsky challenge the assumptions made about the two approaches. An extract is presented in the side note 6.2.

Marvin and Papert's statement is from a period where the goal was to build a general intelligence, not assist humans in their activities. However, I perceive many similarities concerning the choice of models in IML and IMT, which should not exclude DL per se. Future ML research in transfer learning and lightweight DL might provide efficient models to use in interactive systems.

I also see historical analogies between IMT and *expert systems*, which developed in the 1980s on LISP machines (Figure 6.3). Theorized by Edward Feigenbaum, *expert systems* embed a knowledge base and an inference engine. AI researchers designed expert systems to be understood, reviewed and edited by domain experts rather than IT experts. The inference engine applies the rules to the known facts to deduce new facts. The development of expert systems is a key moment in AI history because it is the first time that people other than researchers have appropriated an AI technology. A machine can learn from a person's knowledge if it can be translated into rules and data. Domain-expert can program expert systems themselves with little expertise in programming or with the help of a computer scientist. Furthermore, it is the first time that AI systems are deployed outside of AI research labs, into companies, hospitals, or other research labs from other disciplines [Buchanan and Shortliffe, 1984, Feigenbaum et al., 1970].

In the late 1980s, Personal Computers were more affordable and powerful than the specialized LISP Machines [Markoff]. Consequently,

«Too many people too often speak as though the strategies of thought fall naturally into two groups whose attributes seem diametrically opposed in character:

Symbolic	***Connectionist***
Logical	*Analogical*
Serial	*Parallel*
Discrete	*Continuous*
Localized	*Distributed*
Hierarchical	*Heterarchical*
Left-brained	*Right-brained*

This broad division makes no sense to us, because these attributes are largely independent of one another.»

Later in the preface, the authors claimed that:
«It is just as clear to us today as it was 20 years ago that the marvelous abilities of the human brain must emerge from the parallel activity of vast assemblies of interconnected nerve cells. But, as we explain in our epilogue, the marvelous powers of the brain emerge not from any single, uniformly structured connectionist network but from highly evolved arrangements of smaller and specialized networks which are interconnected in very specific ways.»

Figure 6.2: Extract of the preface of the expanded version of *Perceptron: introduction to computational geometry* [Minsky and Papert, 1969]

the *LISP machine* market collapsed and dragged expert systems down with it. Fundamentally, expert systems presented limitations when the knowledge base became too large. They were not flexible enough to update the knowledge and the inference process became intractable. In their system PICL, Ramos et al. [Ramos et al., 2020] reported possible limitations with high number of samples, features and models. These limitations were attributed to the interface design but future research should investigate if the teaching process itself would become intractable or hard to update with an increasing number of samples, semantic features and models.

Figure 6.3: An example of *LISP machine*, vector of the *expert systems* diffusion in the 1980s. The model shown is a Symbolics 3640.

Furthermore, semantic features and deep learning are not confrontational since DL models can learn semantic features, especially within specific layers of their architecture. In DL, the model takes charge of the feature crafting rather than the user. Beyond automation, DL can be seen as a way to learn intermediate representations. The ML research suggests that users could fix explicit constraints in the way these features are learned in DL models through conditioning and disentanglement [Ridgeway and Mozer, Bengio et al.]. This shift back to a more symbolic deep learning rather than performance-based and end-to-end processing is an opportunity for HCI research to provide new interaction and visualization techniques that offer more agency to DL models parameters. For instance, Boggust et al. [Boggust et al., 2022] designed an embedding comparator that can quickly reveal semantic changes after fine-tuning a model on a new natural language processing task.

If user-crafted semantic features work well with text documents, it is less true with images. Defining a semantic feature with images is challenging. Only deep learning can create abstract and variable semantic features such as “animal” or “sadness”. Image-processing techniques might also be incomprehensible and impossible to manipulate for users for less abstract features. For example, the semantic feature “redness” might be the average of the red pixels in an image or the averaged red pixels minus the averaged green pixels of an image. In the latter case, it might not be evident that greenness is opposed to redness. A feature “redness” might also be confusing if applied to the entire image rather than the object of interest.

I would argue that a modular and semantic vision of IMT models can coexist with deep learning, especially when applied with data in which semantic features are difficult to obtain without deep learning. Deep learning becomes necessary (although challenging) if we consider extending IMT for subject-matter experts like artists, which need

generative models.

6.5 Interactive machine teaching as a tool for ML and AI education

The primary goal of IMT, as defined by Simard et al. [Simard et al., 2017] and Ramos et al. [Ramos et al., 2020], is to support people in the creation of ML models. This thesis suggest that another underlying approach would be to use IMT as a tool to support peoples' literacy about ML and AI.

With a model-building goal, IMT systems that guide users toward an optimal teaching strategy are beneficial. Wall et al. [Wall et al., 2019] endorse this approach with guidance that supports novices to be *"be quickly on-boarded"* with an efficient teaching strategy.

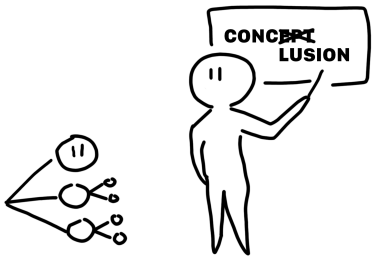
I would argue that there are pedagogical benefits to letting novices explore either good or bad teaching strategies. Retrospectively, several participants were critical of the strategy employed during our studies. In the user experiment presented in section 4.4, some participants drastically changed their teaching strategy from one condition to another. Thus, I would argue that offering people the condition to adopt an exploratory approach to machine teaching (e.g. a badly trained model should not be penalized) can be beneficial to engage them in an investigative mindset that involves self-reflection (i.e. updates on the curriculum and decision-making rules). On the opposite, constraining people with a more fixed teaching strategy and curriculum, either by providing guidance or encouraging them to choose and stick to a single teaching strategy, might be beneficial when machine teachers should efficiently build an accurate ML model.

These considerations open interesting research perspectives regarding usual IMT goals oriented toward model-building. IMT could also aim at designing interactions that could leverage teachers' curiosity or understanding of ML models, either from a functional (model behavior) or structural (model inner working) point of view. More generally, we see in this approach of interactive machine teaching an interesting means for research in ML democratization and education. The work presented in this paper has been initiated through a collaboration with the association *Traces*, dedicated to science popularization. Our collaborators from the association saw the idea of teaching a machine as a means to give people a tool to learn about ML, reflect on it and democ-

ratize it. This idea is gaining a very recent interest in the field of HCI, and CSCW [Hitron et al., 2019, Fiebrink, 2019, Zimmermann-Niefield et al., 2019b, Lee et al., 2019]. Our work is in line with this work, promoting learning, appropriation and decentralized governance of technology and extends it by allowing novice users to be engaged with the expressive capacities of modern ML (deep learning), which means the possibility to convey increasingly rich concepts through data.

Chapter 7

Conclusion



In this thesis, I explore how non-experts users behave when placed in the role of machine teachers i.e. in control of an interactive ML system. I was particularly interested in eliciting their reasoning, strategies, and learning when teaching an IML system. Through two artistic collaborations, I also offer a first-person perspective on the challenges and opportunities of IMT for art.

This thesis is anchored in science popularization collaborations, which allowed us to conduct remote workshops and a user study. The latter is inspired by a structured observation method and uses a think-aloud protocol. It involves participants in teaching an image classifier using sketches they create. Remote participants used Marcelle-Sketch, a sketch recognition application we designed to be incrementally teachable and usable in a web browser. I found that participants engage in heterogeneous teaching strategies regarding sequencing and variability. The variability tends to favor the model generalization abilities, but the type of variability, and the fact it might be introduced progressively, plays a role in building an efficient classifier. Participants discovered new insights about the system by investigating transformations on existing representations and were confused about four inherent neural network properties. These insights contributed to discussing novices' teaching strategies and understanding and the use of IMT as a tool for active pedagogy that can leverage peoples' literacy in ML and AI.

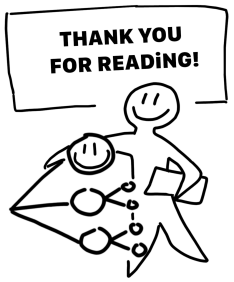
The thesis then explores the place of uncertainty in IMT. In particular, we explored two types of uncertainty in deep learning. From a benchmark analysis investigating transfer learning techniques to perform real-time uncertainty estimation, we found that the variability in the data in the feature space is essential for detecting uncertain

images. We conducted a hybrid controlled experiment using a think-aloud protocol and quantitative tests to probe non-experts' perception and use of uncertainty in an IMT scenario. We found that participants' choices made while teaching—especially regarding training set size and variability—are more important than the type of uncertainty participants were exposed to. We also identified and discussed two teaching approaches: the first uses uncertainty to guide the selection of training data; the second systematically introduces variation across the classes. We found systematic teaching strategies resulted in a better understanding of the classifier outcome. We compared the accuracy of our participants' classifiers with models trained using an active learning procedure from these results. Participants obtained better performances than simulated curricula using AL. All these results fueled the discussion on the utility of a two-level uncertainty in IMT and design directions to support novices in a machine teaching task.

I took a reflective perspective on my involvement in two art-science projects using ML, "Figures dissidentes" and "Cor Epiglottae". Artistic practices raise significant challenges to IMT, such as subjective assessment criteria f models' outcomes, the difficulty of efficiently exploring various generative models that convey different aesthetics, or the unusual situations in which learning systems are deployed (e.g. in interaction with the audience). At the same time, these challenges offer exciting research opportunities such as understanding artists' empirical tricks to control generative models and designing IMT tools dedicated to artistic practices.

Lastly, I discuss the place of modern ML (deep learning) in IMT, highlighting promising prospects in creating interaction and visualization techniques to foster non-experts' agency on the self-taught semantic features of neural networks.

This thesis contributes to seeing machine learning as a human activity that IMT systems can democratize among novices, opening a new perspective to see IMT as a tool for education or artistic creation.



Appendix A

Transfer learning: improving efficacy-expressivity trade-off for the design of more teachable systems using deep learning

Deep Learning presents important technical constraints for their use in IMT systems:

(C1) DNNs usually require thousands to millions of examples to learn a task. Such data is collected once and used as a batch to train the deep learning architecture.

(C2) Similarly, DNNs require a large number of optimization steps to converge.

(C3) At training time, a DNN optimization step is computationally costly because the gradient descent requires to propagate the error across all the neuron layers. This calculation can however be parallelized and computed on Graphics Processing Units (GPU).

For these reasons, DNNs are difficult to embed in interactive ML applications involving rapid and reactive interaction between users and model. This section is intended for an non-expert audience in ML and presents transfer learning techniques that can leverage the constrains above and foster the use of expressive DNNs in interactive and teachable systems.

A.1 Definition

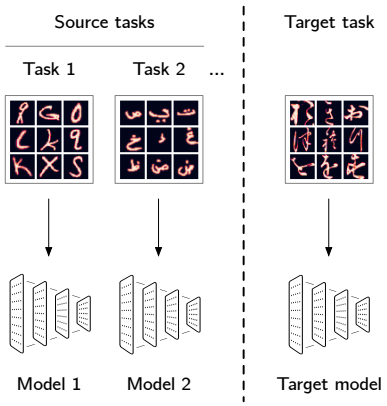
Machine Learning (ML) algorithms are designed to address a single task on which they must minimize an error function. *Transfer learning* focuses on the improvement of an ML algorithm on a new task by using knowledge taken from previous tasks that have already been learned. This approach is illustrated in figure A.1. Transfer learning is valuable for our problem because it can considerably reduce data and computational costs associated with DNNs training compared to the traditional ML approach. More precisely, it can leverage the constraints listed above.

Transfer learning encompasses many terminologies that may confuse the reader. This section does not attempt to be exhaustive but tries to clarify the transfer learning field to a non-ML-expert audience.

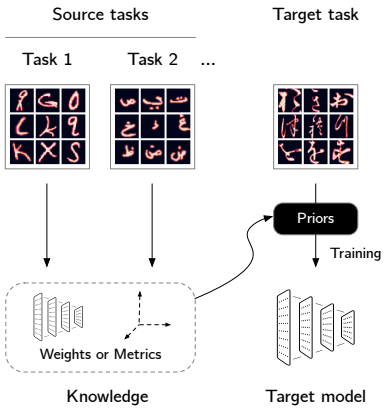
As a first example, *few-shot learning* is an overlapping field of *transfer learning* that aims at building models capable of adapting with very few examples, usually under 20 [Wang et al., 2020]. Some of the transfer techniques presented below also apply to the problem of *few-shot learning*. Throughout this appendix, I try to clarify the expected magnitude of the training efficiency gain for each transfer learning technique.

Aside from the few-shot learning scenario, the rest of the transfer learning literature generally focuses on reaching a good accuracy after the knowledge transfer. The purpose is generally to facilitate training by minimizing the number of examples given or the number of optimization epochs performed. Unlike few-shot learning, minor concerns are placed on reaching drastic training efficiency using few examples. Another branch called distant domain transfer learning focuses on estimating the distance between two tasks and transferring knowledge between very dissimilar tasks.

Transfer learning also encompasses several scenarios that should not be confused, although they are sometimes used interchangeably in the literature. In particular, it is important to distinguish *domain adaptation* or transductive transfer learning from the rest. *Domain adaptation* seeks to adapt to new distributions in the input domain. For example, after training a model to recognize the Latin alphabet with lowercase characters, we want to recognize uppercase characters. The source and target tasks are identical: we want to classify the 26 letters of the Latin alphabet. Only the inputs are different.



(a) Traditional ML approach



(b) Transfer Learning approach

Figure A.1: The traditional ML approach (a) retrain a new model for each new task. The *Transfer Learning* approach (b) tries to extract knowledge from previous related tasks to train a model faster on a new target task.

By contrast, *inductive Transfer learning* seeks to adapt to a new task, no matter if the input distribution is similar or not. For example, after training a model on handwritten characters in the Latin alphabet, we want to recognize Japanese characters. The task is different because the nature and number of predicted outcomes changed. The different scenarios are summarized in figure A.2.

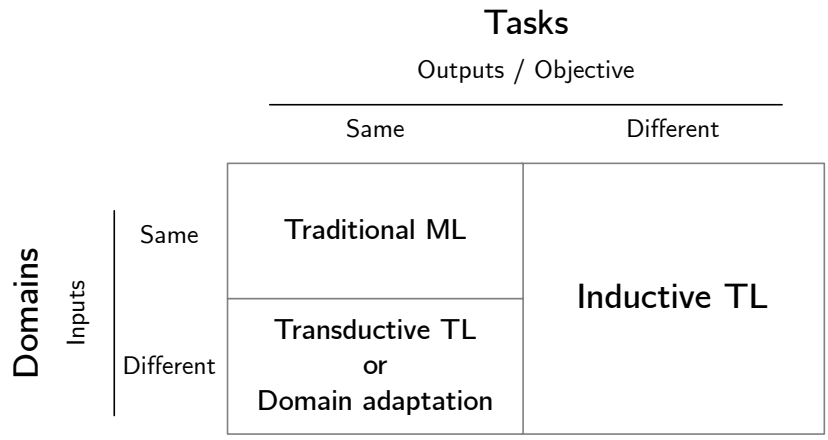


Figure A.2. The transfer learning scenario depends on the similarity or difference between source domains and target domains (inputs) resp. source tasks and target tasks (ouputs and objectives)

Domain adaptation can occur in the online learning scenario, in which the model must learn from examples arriving sequentially. This sequential data source can be non-stationary i.e. the distribution of the incoming inputs is changing over time. Such scenario is called *concept drift*. Webb and colleagues [Webb et al., 2016] proposed taxonomy and formal definitions of the different drifts that might differ on their subject (class drift, covariate drift, novel class appearance), duration, magnitude, or reoccurrence.

Models that are intentionally designed to adapt to both *concept drifts* or task changes are relevant for the design of teachable interactive systems. However, this thesis mainly focuses on *inductive transfer learning* to obtain models that can be taught quickly and with fewer examples. The research in *inductive transfer learning*[1] spans different branches that explored transfer learning with neural networks: **weight transfer**, **deep metric learning**, and **meta-learning**.

[1] We will use only transfer learning for the rest of the manuscript for ease of reading.

A.2 Weight transfer

Transfer learning using weight transfer considers a trained neural network on a source task as an initialization point for training on the target task [Amiriparian et al., 2017, Long et al., Zhang et al., 2017,

Pratt et al.].

Research has shown that the layers of a neural network have different degrees of generality or specificity to a given problem [Yosinski et al., 2014]. For instance, the first layers of convolutional neural networks trained on natural images are systematically responsive to Gabor filters and color blob patterns, as shown in figure A.3 taken from Brachmann and colleagues [Brachmann and Redies, 2016]. Yosinsky and colleagues [Yosinski et al., 2015] developed an interactive visualization tool to explore neurons activation in convolutional networks. Their demo video sheds light on how the deeper layers of an AlexNet convolutional neural network learn specialized features[2]. For example, the fifth convolutional layer is composed of neurons that fire i response to face and shoulders, wrinkles on shirts, or printed text given as input images.

[2] Available in their blog post https://yosinski.com/deepvis

The weight transfer approach does not reuse a pretrained model as is. Source and target models can sometimes be trained simultaneously [Rozantsev et al., Caruana, 1997]. More often, layers are split in two groups to keep the trained generalist layers and only retrain the specialized layers. We consider two groups of layers:

- **The first n layers** that learned general features on the source task. These layers can be either *frozen* or *fine-tuned* when retraining on the target task. *Fine-tuning* implies backpropagating the errors from the new task into the copied source features. By contrast, *freezing* correspond that the transferred feature layers will not change during training on the new target task. We illustrate the *frozen* transfer learning approach in figure A.4. The fine-tuned approach is similar except that pretrained layers are trainable. Choosing to freeze or fine-tune the source neuron layers relies on the size of the target dataset and the number of parameters in the n first layers. It is preferable to leave the features frozen if the target dataset is small and the number of parameters in the n first layers is large. By contrast, if the target dataset is large or the number of parameters is small, we tend to *fine-tune* the neural network.

- **The last layers** ($n+1$ to the end) that learned specialized features on the source task. The neuron weights of these layers are usually reinitialized to be retrained on the target task.

The *frozen* transfer learning approach mitigates constraints (C1), (C2) and (C3) mentioned above since a smaller model is trained. The *fine-tuned* approach only affect (C1) and (C2).

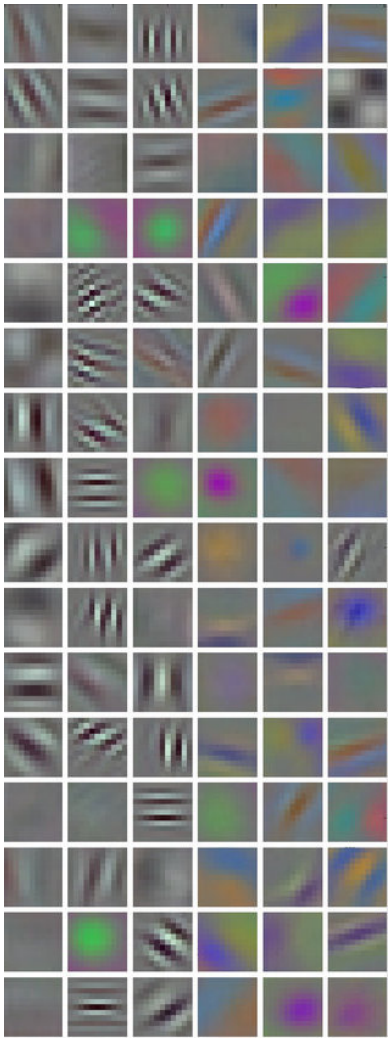


Figure A.3: Gabor filter and color blobs patterns systematically activate neurons in the first layer of a convolutional neural networks. Figure taken from [Brachmann and Redies, 2016]

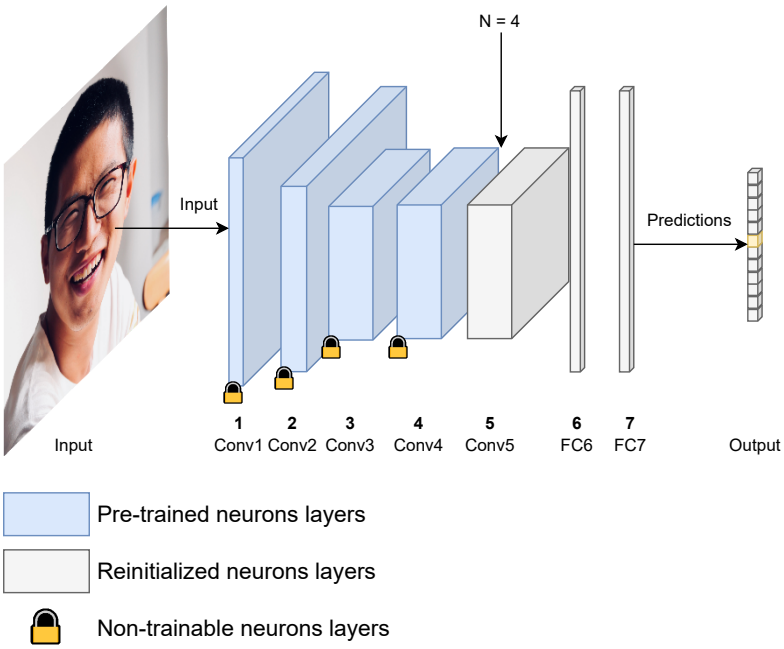


Figure A.4. Weight transfer in deep neural networks using frozen pretrained neurons layers. The fine-tuning approach does not freeze the pretrained neurons layers.

The question is now about the choice of n i.e. to know on which layer we should split the neural network in two. Yolinski and colleagues [Yosinski et al., 2014] conducted systematic investigations on this problem using the popular AlexNet convolutional neural network composed of 7 hidden layers. They explored transfer learning varying n from 0 (retrain all the model, not transfer learning) to 7 (the pretrained model is used as is) both with *freezing* and *fine-tuning* the first layers. Their results first showed that transfer learning is negatively affected by the specialization of higher layers to the source task at the expense of the target task. Thus, performing weight transfer on the last layers is often contraindicated. Second, they demonstrated that consecutive layers in the middle of the neural network might be co-adapted, and splitting between two co-adapted layers might result in a poor transfer. Overall, the transfer was performed on large target datasets ($k > 1000$) and observed a modest improvement in accuracy over the baseline condition on ignoring the source task and retraining the full network on the target task from scratch. However, weight transfer yields considerable calculation savings. Weight transfer was also applied to domain adaptation [Oquab et al., Rozantsev et al.].

A.3 Meta-learning

We saw that weight transfer focus on reusing trained weights from source task to target task [Vanschoren, 2018, Vilalta and Drissi, 2002, Li et al., Hsu et al., 2019]. Meta-learning aims at building models that learn to learn. In this paradigm, researchers not only consider a single source task but a variety of tasks on which the model is trained to adapt from one to another, as illustrated in figure A.5. The pretrained model is generally trained to adapt across the different tasks. After this training procedure, the resulting weights are expected to quickly adapt to a new related task with only a few training examples.

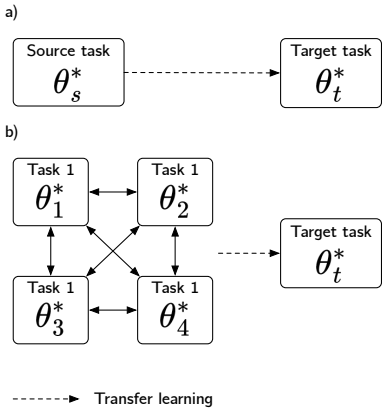


Figure A.5: (a) Traditional transfer learning techniques transfer knowledge from a single source to a new target task. (b) Meta-learning techniques developed by [Finn et al., 2017] consider an ensemble of tasks on which the model is trained to be able to adapt from one to the other. After this meta-training process, the resulting weights can be efficiently fine-tuned on a new target class.

Finn and colleagues [Finn et al., 2017] proposed a popular model-agnostic approach that trains a model on several tasks at each optimization epoch and enables to do few-shot learning. At each epoch, the meta-model weights are the averaged weights of all fine-tuned models on each task. This training process leads to sub-optimal performance on all tasks. However, few optimization epochs on a new related task lead to fast adaptation and performance improvements. This approach responds to the constraints (C1) and (C2) listed above and can be used for few-shot learning. The training procedure is illustrated on figure A.7.

This popular meta-learning approach presents two drawbacks. First, model agnostic meta-learning complexifies data collection since the data must comprise multiple labeled tasks. Second, the meta-training process is computationally expensive, but more efficient algorithms using first-order approximations were proposed and exhibit similar performances [Nichol et al., 2018]. As an example, the authors of the Reptile meta-learning algorithm demonstrate their algorithm with a one-shot interactive sketch-based classifier on their blog[3]. A screenshot of the toy application is presented below in Figure A.6.

[3] https://openai.com/blog/reptile/

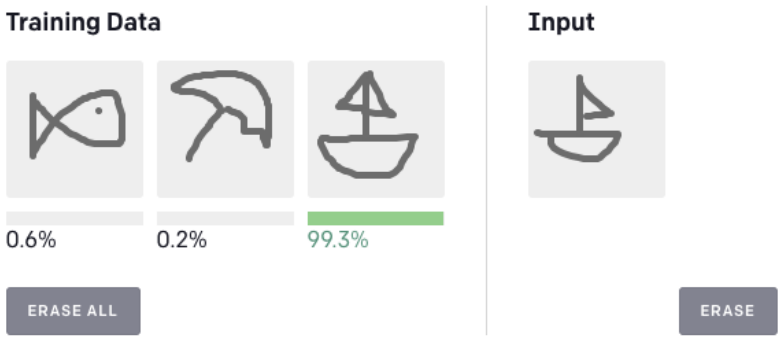


Figure A.6. One-shot learning application using Reptile.

A.4 Deep metric learning

The deep metric learning approach aims at creating *embedding* which is a feature space suited to a large set of tasks, including both the source and target tasks [Scott et al., 2018, Bellet et al., 2013, Chopra et al., Schroff et al.]. In other words, deep metric learning learns transferable features for a set of problems. The embeddings are then used as features to train a shallow neural network or even a simple distance-based model such as k Nearest Neighbors, enabling few-shot learning. The deep metric learning approach reuses the first layers of a pre-trained network, which are generally trained on vast datasets with a large number of classes. These pre-trained layers can also be further optimized using a metric-learning loss, which ensures that instances from the same class are close to each other and distant from instances from other classes [Ustinova and Lempitsky, Ridgeway and Mozer, Wang et al.]. Scott and colleagues showed that *histogram loss* [Ustinova and Lempitsky] is the state-of-the-art metrics in deep metric learning. MobileNet architectures [Howard et al., 2017, Sandler et al., 2018a, Koonce, 2021] are popular models for creating embeddings in computer vision. They are designed with depth-wise separable convolutions to build lightweight models on which the trade-off between accuracy and latency can be adjusted using a single hyper-parameter. Several few-shot learning techniques such as prototypical learning [Snell et al.] or matching networks [Vinyals et al.] rely on well-suited embeddings to the target task. Natural language processing also relies on embedding, such as the popular word2vec embedding [Goldberg et al., 2014].

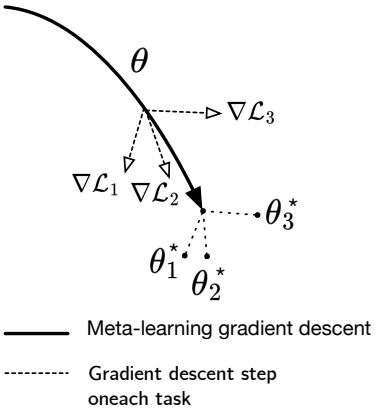


Figure A.7: Diagram of the model-agnostic meta-learning algorithm (MAML) [Finn et al., 2017]. The training process optimizes a representation that can quickly adapt to new tasks. The parameters represented by the point at the extremity of the plain arrow are closed to optimal parameters θ_i^* on each task.

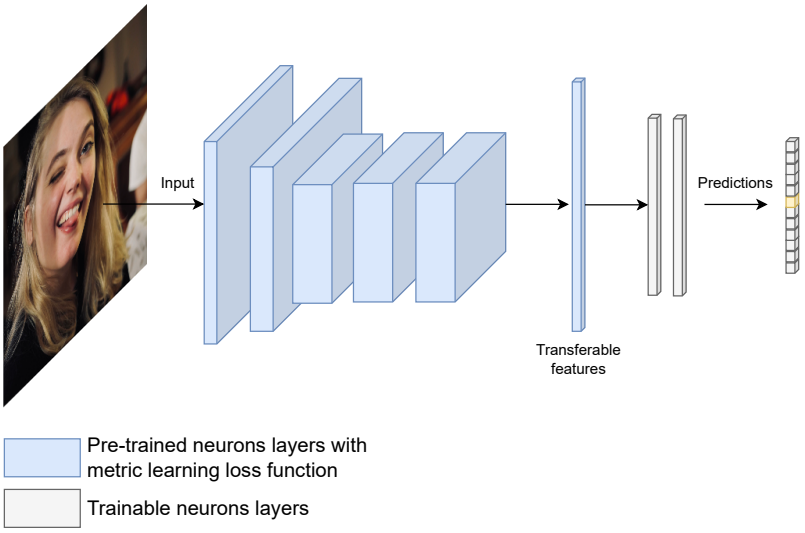


Figure A.8. Illustration of the transfer learning using the deep metric learning approach.

Overall, deep embeddings are much more efficient than weight transfer for designing models that can adapt with few examples. Meta-learning can also be used for few-shot learning but requires retraining the entire model for fine-tuning. The pretraining procedure is also more demanding in terms of data preprocessing.

Research is still active on understanding how far knowledge can be transferred on dissimilar tasks. Furthermore, these techniques permit us to take advantage of both the expressiveness of deep neural networks and the rapid adaptation of shallow models. If the benefits on the training efficiency are impressive, it is not clear how priors prevent high model specialization. For example, an embedding trained on 1000 classes such as MobileNetV1 might not work when used as features for retraining a model on a binary task with medical chest scans images in which the changes for detecting a disease might be very subtle.

Appendix B

Data acquisition scenarios in active learning

Active learning spans three data acquisition scenarios, which comprise specific techniques.

Pool-based scenario [Lewis et al., 1994] is the most common scenario in which data is available all at once. The model can pick a query from the unlabeled pool of data. This approach is illustrated in figure B.1.

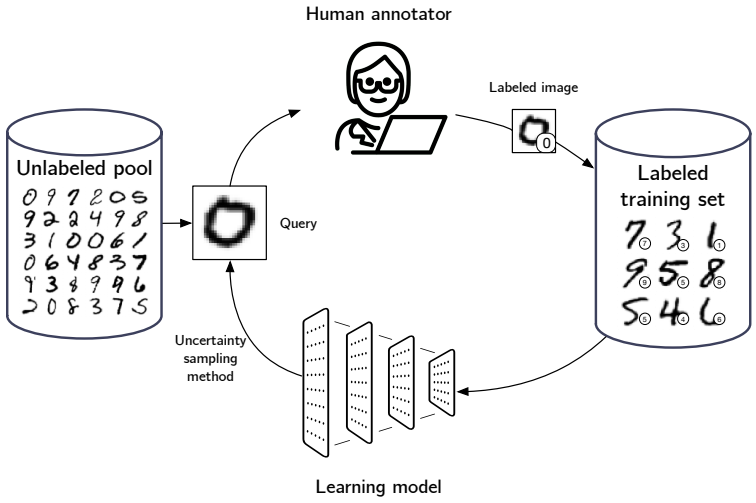


Figure B.1. Pool-based active learning using uncertainty as sampling method.

Stream-based scenario: By contrast to pool-based AL, stream-based AL "makes immediate query decisions at each instance during a single scan of the data stream" [Loy et al., 2012]. At each sample within a sequence, the algorithm decides whether to query a label or discard this sample. This approach does not require computations across a large

unlabeled data set but supposes that sream data acquisition is cheap. Stream-based AL aso ignores the underlying data distribution [Ho and Wechsler, 2008] which is prone to concept evolution [Mohamad et al., 2018, Zliobaite et al., 2014, Loy et al., 2012]. This approach is illustrated in figure B.2.

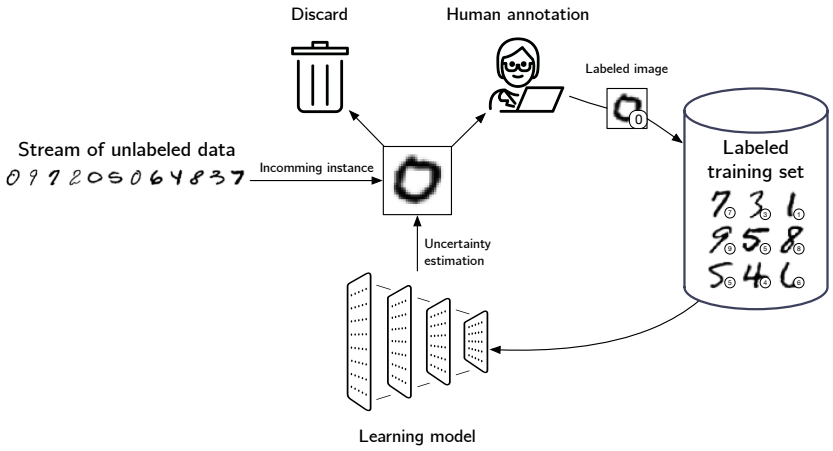


Figure B.2. Stream-based active learning using uncertainty as a decision method.

Membership Query Synthesis [Angluin, 1988] considers that models can query for any unlabeled instance in input space, including queries that the model generates de novo, rather than those sampled from some underlying natural distribution [Settles, 2010].

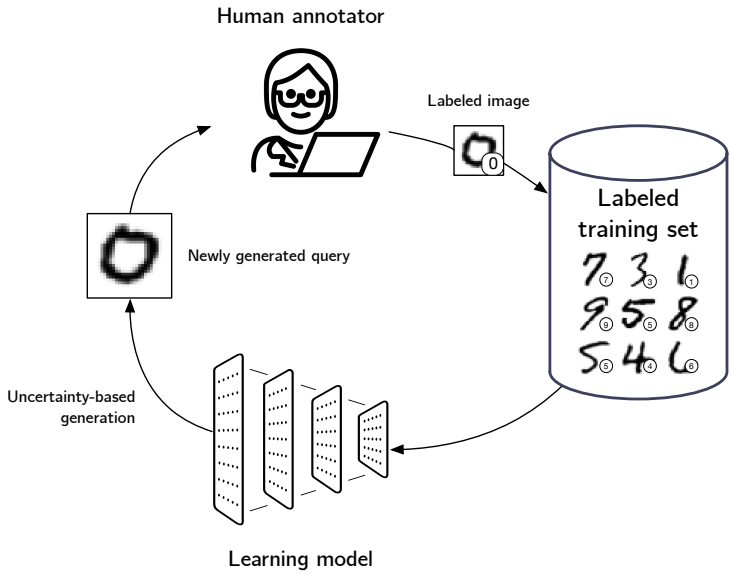


Figure B.3. Membership Query Sythesis AL, that generates an uncertain query de novo.

Appendix C

Aesthetics of mode-covering or mode-seeking generative ML models

Generative deep neural networks are increasingly used in art since 2016. In particular, Generative Adversarial Network (GAN) become popular models in visual art because they can successfully generate realistic images. GANs are composed of two sub-models. The first model is a generator, which aims to generate realistic images. The second model is a discriminator trained to distinguish a real dataset image from fake images created by the generator. Both models are trained simultaneously i.e. the generator becomes better and better at generating realistic images while the discriminator becomes better and better at distinguishing fake images. Figure C.1 illustrate the training procedure of a GAN architecture. The progress balance between the

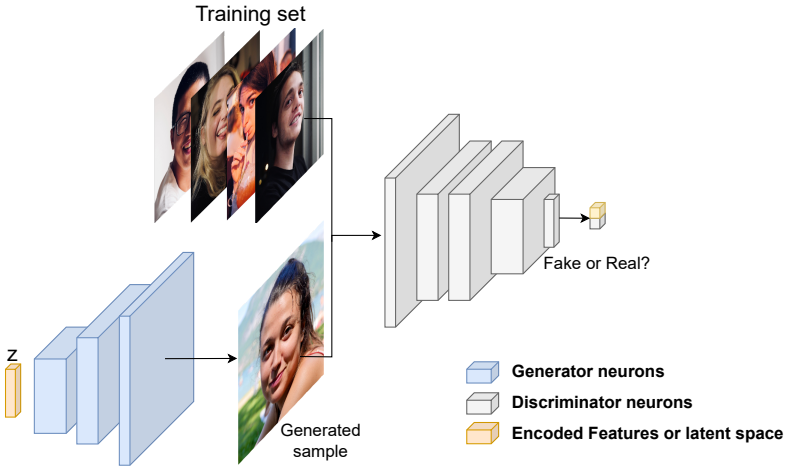


Figure C.1. Schema of a Generative Adversarial Networks (GAN) architecture

generator and the discriminator is key for training a GAN that can

generate a great variety of samples. In many situations, parameters oscillate and never converge, or the discriminator gets too successful compared to the generator leading to a limited variety of generations.

A less popular generative model is the VAE, which is a probabilistic model that learns the conditional dependence structure between random variables, the original image, and its reconstruction.

It turns out that VAEs and GANs outcome distinctive aesthetics. Figure C.2 [Larsen et al., 2016] compare generations from a VAE or a GAN. It shows the distinctive aesthetic of images generated with a GAN decoder compared to a VAE decoder. GANs are known to produce more realistic images with exciting textures and artifacts. VAE results in a more blurry and spectral aesthetic and is more difficult to scale with higher resolution images.

This difference can be explained by the fact that VAE are **mode-covering** while GANs are **mode-seeking**. The difference between mode-covering and mode-seeking lies in the compromises made when a model does not have enough capacity to capture all the variability in the data. Likelihood-based models such as VAEs are mode-covering i.e. they overgeneralize and produce interpolations that may not be meaningful. This is due to the maximization of the joint likelihood of the data. Adversarial models such as GANs are mode-seeking because the loss can be minimized without necessarily trying to reproduce all datapoints characteristics, as long as the produced images trick the discriminator. Thus, some parts of the distribution are ignored.

As illustrated above in Figure C.2, GANs provide a sharply different aesthetic than VAEs, with their own distinctive and recognizable textures and artifacts. Researchers and artists ironically invented the term GANism as a modern art movement, echoing famous painting movements (impressionism, cubism, fauvism etc.). In a Tweet from 2017, the ML engineer and developer François Chollet claimed that *«GANism (the specific look and feel of seemingly GAN-generated images) may yet become a significant modern art trend»*.

Thus, if likelihood-based models homogennize the images generated by design, GANs also have a uniform and recognizable aesthetic that artists might want to challenge and escape from to offer novel artistic productions [1]. Indeed, GANs are already quite old (2016) but have diversified. New generative models use text-guided diffusion models [Dhariwal and Nichol, Nichol et al., Kim and Ye, 2021]. Taken from a publicly available Git repository or online platform (Google Collab),

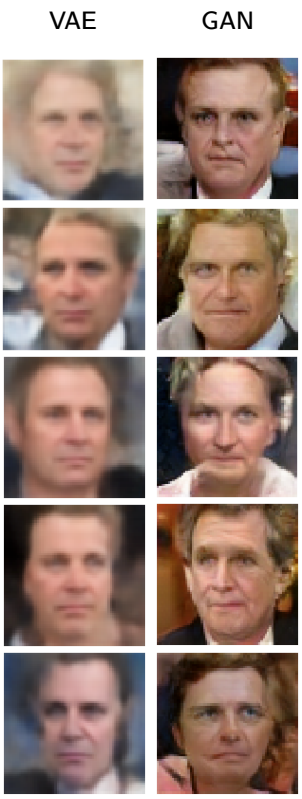


Figure C.2: Samples generated by a VAE (left) or a GAN (right) from a single encoded image. Images taken from Larsen and colleagues [Larsen et al., 2016]

[1] In a tweet from 2021, the pioneer artist of ML-art Mario Klingemann claimed: *«I do not really care much for "pretty" generative art. I want something that has an interesting concept, is algorithmically challenging and ideally so complex that I cannot reverse engineer its mechanism right away. Unfortunately I don't see a lot of that these days.»*

models might convey a uniform aesthetic that artists might need to tweak and escape from. To do so, artists must develop transverse expertise across different models and develop their workflow by chaining or composing different models to create novel pieces.

Bibliography

Most.js: Monadic event stream. https://github.com/mostjs/core/. (Accessed on 04/07/2021).

Compas (software), 2004a. URL https://en.wikipedia.org/wiki/COMPAS_(software).

Facial recognition system, 2004b. URL https://en.wikipedia.org/w/index.php?title=Facial_recognition_system.

Bringing ai into the classroom, 2019. URL https://www.actua.ca/en/bringing-ai-into-the-classroom.

M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, M. Kudlur, J. Levenberg, R. Monga, S. Moore, D. G. Murray, B. Steiner, P. Tucker, V. Vasudevan, P. Warden, M. Wicke, Y. Yu, and X. Zheng. Tensorflow: A system for large-scale machine learning. In *12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16)*, pages 265–283, Savannah, GA, Nov. 2016. USENIX Association. ISBN 978-1-931971-33-1. URL https://www.usenix.org/conference/osdi16/technical-sessions/presentation/abadi.

A. Agassi, H. Erel, I. Wald, and O. Zuckerman. Scratch nodes ML: A playful system for children to create gesture recognition classifiers. *dl.acm.org*, 5 2019. DOI: 10.1145/3290607.3312894. URL https://dl.acm.org/doi/abs/10.1145/3290607.3312894?casa_token=9HfhBOcZTpMAAAAA:Lx8mZRCe1qUTm2EdzHg5v4kEIUaL4p2f05Ljn2doXciNyDT825lvzIPQk3RncPYMFdunZ9Bp-NDckA.

S. Amershi, M. Cakmak, W. Knox, and T. Kulesza. Power to the people: The role of humans in interactive machine learning. *ojs.aaai.org*. URL https://ojs.aaai.org/index.php/aimagazine/article/view/2513.

S. Amershi, J. Fogarty, A. Kapoor, and D. Tan. Designing for effective end-user interaction with machine learning. *UIST'11 Adjunct - Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology*, pages 47–50, 2011. DOI: 10.1145/2046396.2046416. URL https://ojs.aaai.org/index.php/AAAI/article/view/7964.

S. Amershi, M. Chickering, S. M. Drucker, B. Lee, P. Simard, and J. Suh. Modeltracker: Redesigning performance analysis tools for machine learning. *Conference on Human Factors in Computing Systems - Proceedings*, 2015-April:337–346, 4 2015. DOI: 10.1145/2702123.2702509.

S. Amershi, D. Weld, M. Vorvoreanu, A. Fourney, B. Nushi, P. Collisson, J. Suh, S. Iqbal, P. N. Bennett, K. Inkpen, et al. Guidelines for human-ai interaction. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, pages 1–13, 2019.

S. Amiriparian, M. Gerczuk, S. Ottl, N. Cummins, M. Freitag, S. Pugachevskiy, A. Baird, and B. Schuller. Snore sound classification using image-based deep spectrum features. 2017. DOI: 10.21437/Interspeech.2017-434. URL https://opus.bibliothek.uni-augsburg.de/opus4/files/65822/0434.PDF.

D. Angluin. Queries and Concept Learning. *Machine Learning*, 2(4):319–342, 1988. ISSN 15730565. DOI: 10.1023/A:1022821128753. URL http://scholar.google.com/scholar?hl=en&btnG=Search&q=intitle:Queries+and+concept+leaming#0.

J. Attenberg and F. Provost. Why label when you can search? Alternatives to active learning for applying human resources to build classification models under extreme class imbalance. *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 423–432, 2010. DOI: 10.1145/1835804.1835859. URL http://www.adsafemedia.com.

J. Attenberg, P. Ipeirotis, and F. Provost. Beat the machine: Challenging workers to find the unknown unknowns. *aaai.org*. URL https://www.aaai.org/ocs/index.php/WS/AAAIW11/paper/viewPaper/3954.

A. Baevski, W. Hsu, Q. Xu, A. Babu, and J. Gu. Data2vec: A general framework for self-supervised learning in speech, vision and language. *arxiv.org*. URL https://arxiv.org/abs/2202.03555.

E. Bainomugisha, A. L. Carreton, T. v. Cutsem, S. Mostinckx, and W. d. Meuter. A survey on reactive programming. *ACM Computing Surveys*, 45(4), Aug. 2013. ISSN 0360-0300. DOI: 10.1145/2501654.2501666. URL https://doi.org/10.1145/2501654.2501666.

M. Bakator and D. Radosav. Deep learning and medical diagnosis: A review of literature. *mdpi.com*, 2018. DOI: 10.3390/mti2030047. URL https://www.mdpi.com/328462.

A. Bellet, A. Habrard, and M. Sebban. A Survey on Metric Learning for Feature Vectors and Structured Data. 6 2013. URL http://arxiv.org/abs/1306.6709.

W. H. Beluch, T. Genewein, A. Nürnberger, and J. M. Köhler. The Power of Ensembles for Active Learning in Image Classification. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 9368–9377, 2018. ISSN 10636919. DOI: 10.1109/CVPR.2018.00976.

Y. Bengio, A. Courville, and P. Vincent. Representation Learning: A Review and New Perspectives. URL http://www.image-net.org/challenges/LSVRC/2012/results.html.

J. J. Benjamin, A. Berger, N. Merrill, and J. Pierce. Machine Learning Uncertainty as a Design Material: A Post-Phenomenological Inquiry. 2021. DOI: 10.1145/3411764.3445481. URL http://arxiv.org/abs/2101.04035%0Ahttp://dx.doi.org/10.1145/3411764.3445481.

S. Berg, D. Kutra, T. Kroeger, C. N. Straehle, B. X. Kausler, C. Haubold, M. Schiegg, J. Ales, T. Beier, M. Rudy, K. Eren, J. I. Cervantes, B. Xu, F. Beuttenmueller, A. Wolny, C. Zhang, U. Koethe, F. A. Hamprecht, and A. Kreshuk. ilastik: interactive machine learning for (bio)image analysis. URL https://forum.image.sc/tags/ilastik.

F. Bernardo, M. Zbyszynski, R. Fiebrink, and M. Grierson. Interactive Machine Learning for End-User Innovation. 2017. URL www.aaai.org.

C. Bhatt, K. Udham Singh, A. Kumar, I. Kumar, and V. Vijayakumar. The state of the art of deep learning models in medical science and their challenges. *Springer*, 27(4):599–613, 8 2021a. DOI: 10.1007/s00530-020-00694-1. URL https://link.springer.com/article/10.1007/s00530-020-00694-1.

U. Bhatt, J. Antorán, Y. Zhang, Q. V. Liao, P. Sattigeri, R. Fogliato, G. Melançon, R. Krishnan, J. Stanley, O. Tickoo, et al. Uncertainty as a form of transparency: Measuring, communicating, and using uncertainty. In *Proceedings of the 2021 AAAI/ACM Conference on AI, Ethics, and Society*, pages 401–413, 2021b.

A. Boggust, B. Carter, and A. Satyanarayan. Embedding Comparator: Visualizing Differences in Global Structure and Local Neighborhoods via Small Multiples. pages 746–766, 3 2022. DOI: 10.1145/3490099.3511122.

A. Brachmann and C. Redies. Using convolutional neural network filters to measure left-right mirror symmetry in images. *Symmetry*, 8(12), 2016. ISSN 20738994. DOI: 10.3390/sym8120144. URL https://www.mdpi.com/168370.

V. Braun and V. Clarke. Using thematic analysis in psychology. *Qualitative research in psychology*, 3(2): 77–101, 2006.

B. Buchanan and E. Shortliffe. Rule-based expert systems: the MYCIN experiments of the Stanford Heuristic Programming Project. 1984. URL http://papers.cumincad.org/cgi-bin/works/Show&_id=caadria2010_044/paper/ec87.

J. Bullock and A. Momeni. Ml.Lib: Robust, Cross-Platform, Open-Source Machine Learning for Max and Pure Data. In *Proceedings of the International Conference on New Interfaces for Musical Expression*, NIME 2015, page 265–270, Baton Rouge, Louisiana, USA, 2015. The School of Music and the Center for Computation and Technology (CCT), Louisiana State University. ISBN 9780692495476.

C. J. Cai, E. Reif, N. Hegde, J. Hipp, B. Kim, D. Smilkov, M. Wattenberg, F. Viegas, G. S. Corrado, M. C. Stumpe, M. Terry, and G. S. Cor-Rado. Human-Centered Tools for Coping with Imperfect Algorithms During Medical Decision-Making. page 14. DOI: 10.1145/3290605.3300234. URL https://doi.org/10.1145/3290605.3300234.

M. Cakmak and A. Thomaz. Designing robot learners that ask good questions. *HRI'12 - Proceedings of the 7th Annual ACM/IEEE International Conference on Human-Robot Interaction*, pages 17–24, 2012a. DOI: 10.1145/2157689.2157693.

M. Cakmak and A. Thomaz. Optimality of human teachers for robot learners. *ieeexplore.ieee.org*, 2012b. URL https://ieeexplore.ieee.org/abstract/document/5578865/.

M. Cakmak, C. Chao, and A. L. Thomaz. Designing Interactions for Robot Active Learners. *IEEE Transactions on Autonomous Mental Development*, 2(2):108–118, 2010. ISSN 1943-0604. DOI: 10.1109/TAMD.2010.2051030.

B. Caramiaux and M. Donnarumma. Artificial Intelligence in Music and Performance: A Subjective Art-Research Inquiry. *Handbook of Artificial Intelligence for Music*, pages 75–95, 2021. DOI: 10.1007/978-3-030-72116-9_4.

B. Caramiaux, J. Françoise, N. Schnell, and F. Bevilacqua. Mapping through listening. *ieeexplore.ieee.org*, 2013. DOI: 10.1162/COMJ. URL https://ieeexplore.ieee.org/abstract/document/6899813/.

B. Caramiaux, M. Donnarumma, and A. Tanaka. Understanding Gesture Expressivity through Muscle Sensing. *ACM Transactions on Computer-Human Interaction*, 21(6):1–26, 2015. ISSN 10730516. DOI: 10.1145/2687922. URL http://dl.acm.org/citation.cfm?doid=2722827.2687922.

D. Cardon, J. P. Cointet, and A. Mazières. Neurons Spike Back. *Reseaux*, 211(5):173–220, 2018. ISSN 07517971. DOI: 10.3917/res.211.0173. URL https://neurovenge.antonomase.fr/NeuronsSpikeBack.pdf.

M. Carney, B. Webster, I. Alvarado, K. Phillips, N. Howell, J. Griffith, J. Jongejan, A. Pitaru, and A. Chen. Teachable machine: Approachable web-based tool for exploring machine learning classification. *Conference on Human Factors in Computing Systems - Proceedings*, 4 2020a. DOI: 10.1145/3334480.3382839.

M. Carney, B. Webster, I. Alvarado, K. Phillips, N. Howell, J. Griffith, J. Jongejan, A. Pitaru, and A. Chen. Teachable machine: Approachable web-based tool for exploring machine learning classification. In *Extended abstracts of the 2020 CHI conference on human factors in computing systems*, pages 1–8, 2020b.

R. Caruana. Multitask Learning. *Machine Learning*, 28(1):41–75, 1997. ISSN 08856125. DOI: 10.1023/A:1007379606734.

R. Caruana, Y. Lou, J. Gehrke, P. Koch, and M. Sturm. Intelligible models for healthcare: Predicting pneumonia risk and hospital 30-day readmission. *dl.acm.org*, 2015-Augus:1721–1730, 8 2015. DOI: 10.1145/2783258.2788613. URL https://dl.acm.org/doi/abs/10.1145/2783258.2788613?casa_token=TjtK2ZbDlb0AAAAA:KfE3mF1qYlquVw1627Ql2R8oK57jsRZURomA7SZMBr7B3aBKpi12SeAI-j9bHR3E6ICkHsCdXSMm.

C. Chao, M. Cakmak, and A. L. Thomaz. Transparent active learning for robots. *ieeexplore.ieee.org*. URL https://ieeexplore.ieee.org/abstract/document/5453178/.

S. Chernova and A. L. Thomaz. Robot learning from human teachers. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, 28:1–121, 4 2014. ISSN 19394616. DOI: 10.2200/S00568ED1V01Y201402AIM028.

S. Chopra, R. Hadsell, and Y. LeCun. Learning a similarity metric discriminatively, with application to face verification. *ieeexplore.ieee.org*. URL https://ieeexplore.ieee.org/abstract/document/1467314/.

N. C. F. Codella, D. Gutman, M. E. Celebi, B. Helba, M. A. Marchetti, S. W. Dusza, A. Kalloo, K. Liopyris, N. Mishra, H. Kittler, and A. Halpern. Skin lesion analysis toward melanoma detection: A challenge at the 2017 International symposium on biomedical imaging (ISBI), hosted by the international skin imaging collaboration (ISIC). In *2018 IEEE 15th International Symposium on Biomedical Imaging (ISBI 2018)*, pages 168–172. IEEE, 4 2018. ISBN 978-1-5386-3636-7. DOI: 10.1109/ISBI.2018.8363547. URL https://ieeexplore.ieee.org/document/8363547/.

D. A. Cohn, Z. Ghahramani, and M. I. Jordan. Active learning with statistical models. *Journal of artificial intelligence research*, 4:129–145, 1996.

T. Cook, D. Campbell, and A. Day. *Quasi-experimentation: Design & analysis issues for field settings*. 1979. URL http://dickyh.staff.ugm.ac.id/wp/wp-content/uploads/2009/ringkasan%20buku%20quasi-experimentakhir.pdf.

E. Delaney, D. Greene, and M. T. Keane. Uncertainty estimation and out-of-distribution detection for counterfactual explanations: Pitfalls and solutions. *arXiv preprint arXiv:2107.09734*, 2021.

T. DeVries and G. W. Taylor. Learning confidence for out-of-distribution detection in neural networks. In *arXiv preprint arXiv:1802.04865*, 2018.

P. Dhariwal and A. Nichol. Diffusion models beat gans on image synthesis. *proceedings.neurips.cc*. URL https://proceedings.neurips.cc/paper/2021/hash/49ad23d1ec9fa4bd8d77d02681df5cfa-Abstract.html.

C. Diaz, P. Perry, and R. Fiebrink. Interactive machine learning for more expressive game interactions. *ieeexplore.ieee.org*. URL https://ieeexplore.ieee.org/abstract/document/8848007/.

F. Doshi-Velez and B. Kim. Towards A Rigorous Science of Interpretable Machine Learning. *arXiv preprint arXiv:1702.08608*, 2017. URL http://arxiv.org/abs/1702.08608.

G. Dove, K. Halskov, J. Forlizzi, and J. Zimmerman. UX design innovation: Challenges for working with machine learning as a design material. *Conference on Human Factors in Computing Systems - Proceedings*, 2017-May:278–288, 5 2017. DOI: 10.1145/3025453.3025739.

J. J. Dudley and P. O. Kristensson. A review of user interface design for interactive machine learning. *ACM Transactions on Interactive Intelligent Systems (TiiS)*, 8(2):1–37, 2018.

U. Dwivedi, J. Gandhi, R. Parikh, M. Coenraad, E. Bonsignore, and H. Kacorri. Exploring Machine Teaching with Children.

M. Faber. On the treatment of uncertainties and probabilities in engineering decision analysis. 2005. URL https://asmedigitalcollection.asme.org/offshoremechanics/article-abstract/127/3/243/468180.

J. A. Fails and D. R. Olsen. Interactive machine learning. *International Conference on Intelligent User Interfaces, Proceedings IUI*, pages 39–45, 2003. DOI: 10.1145/604045.604056.

E. Feigenbaum, B. Buchanan, and J. Lederberg. On generality and problem solving: A case study using the DENDRAL program. 1970. URL https://www.researchgate.net/profile/Bruce-Buchanan/publication/23865744_On_generality_and_problem_solving_A_case_study_using_the_DENDRAL_program/links/0c96052e1e6f7e6a57000000/On-generality-and-problem-solving-A-case-study-using-the-DENDRAL-program.pdf.

Y. Feng, J. Xiao, Z. Zha, H. Zhang, and Y. Yang. Active learning for social image retrieval using Locally Regressive Optimal Design. *Neurocomputing*, 95:54–59, 2012. ISSN 09252312. DOI: 10.1016/j.neucom.2011.06.037.

L. Feugère, C. d'Alessandro, B. Doval, and O. Perrotin. Cantor Digitalis: chironomic parametric synthesis of singing. *Eurasip Journal on Audio, Speech, and Music Processing*, 2017(1), 12 2017. ISSN 16874722. DOI: 10.1186/S13636-016-0098-5.

R. Fiebrink. Machine learning education for artists, musicians, and other creative practitioners. *ACM Transactions on Computing Education (TOCE)*, 19(4):1–32, 2019.

R. Fiebrink and B. Caramiaux. *The machine learning algorithm as creative musical tool*. Oxford University Press, 2018.

R. Fiebrink, D. Trueman, and P. R. Cook. A meta-instrument for interactive, on-the-fly machine learning. 2009. URL https://ualresearchonline.arts.ac.uk/id/eprint/16687/1/FiebrinkTruemanCook_NIME2009.pdf.

R. Fiebrink, P. R. Cook, and D. Trueman. Human model evaluation in interactive supervised learning. *Conference on Human Factors in Computing Systems - Proceedings*, pages 147–156, 2011. DOI: 10.1145/1978942.1978965.

C. Finn, P. Abbeel, S. L. o. m. learning, and u. 2017. Model-agnostic meta-learning for fast adaptation of deep networks. *proceedings.mlr.press*, 2017. URL https://proceedings.mlr.press/v70/finn17a.html.

J. C. Flanagan. The critical incident technique. *Psychological Bulletin*, 51(4):327–358, 1954. ISSN 00332909. DOI: 10.1037/h0061470. URL https://psycnet.apa.org/record/1955-01751-001.

J. Fogarty, D. Tan, A. Kapoor, and S. Winder. CueFlik: Interactive concept learning in image search. *Conference on Human Factors in Computing Systems - Proceedings*, pages 29–38, 2008. DOI: 10.1145/1357054.1357061.

J. Françoise, N. Schnell, R. Borghesi, and F. Bevilacqua. Probabilistic models for designing motion and sound relationships. In *Proceedings of the International Conference on New Interfaces for Musical Expression*, pages 287–292, London, United Kingdom, June 2014. Goldsmiths, University of London. DOI: 10.5281/zenodo.1178764. URL http://www.nime.org/proceedings/2014/nime2014_482.pdf.

G. Franchi, A. Bursuc, E. Aldea, S. Dubuisson, and I. Bloch. TRADI: Tracking deep neural network weight distributions. Technical report, 2020. URL https://hal.archives-ouvertes.fr/hal-02922336.

J. Françoise, B. Caramiaux, and T. Sanchez. Marcelle: Composing Interactive Machine Learning Workflows and Interfaces;. 2021. DOI: 10.1145/3472749.3474734ï. URL https://hal.archives-ouvertes.fr/hal-03335115.

J. Françoise and F. Bevilacqua. Motion-sound mapping through interaction: An approach to user-centered design of auditory feedback using machine learning. *dl.acm.org*, 8(2), 6 2160. DOI: 10.1145/3211826. URL https://dl.acm.org/doi/abs/10.1145/3211826?casa_token=WAynH92oRLYAAAAA:Wp7Po6tu5gptcP0E0SwisDJ1ok92jYgzfm_t8iD7NeJ1Qh63VpuAH4vjyHogt5-dTDWzMpycekyOmg.

Y. Gal. Uncertainty in deep learning. *University of Cambridge*, 1:3, 2016.

Y. Gal and Z. Ghahramani. Dropout as a Bayesian Approximation: Appendix. 6 2015. URL http://arxiv.org/abs/1506.02157.

Y. Gal, R. Islam, and Z. Ghahramani. Deep Bayesian Active Learning with Image Data. *arXiv preprint arXiv:1703.02910*, 2017. URL http://arxiv.org/abs/1703.02910.

J. Garcia, T. Tsandilas, C. Agon, and W. Mackay. Structured observation with polyphony: a multifaceted tool for studying music composition. *dl.acm.org*, pages 199–208, 2014. DOI: 10.1145/2598510.2598512. URL https://dl.acm.org/doi/abs/10.1145/2598510.2598512?casa_token=raYfBuS1INwAAAAA:GQTI5-lbAkvGtWNIBpXo8u1CDmkNWfaho_eTWPDhE16BX7OF1OPZD6q9FVrBb_AHxOqWhAx3gr_87w.

B. Ghai, Q. V. Liao, Y. Zhang, and K. Mueller. Explainable active learning (xal): An empirical study of how local explanations impact annotator experience. *arXiv*, 2020. ISSN 23318422.

M. Gillies, R. Fiebrink, A. Tanaka, J. Garcia, F. Bevilacqua, A. Heloir, F. Nunnari, W. Mackay, S. Amershi, B. Lee, N. D'alessandro, J. Tilmanne, T. Kulesza, and B. Caramiaux. Human-centered machine learning. *Conference on Human Factors in Computing Systems - Proceedings*, 07-12-May-:3558–3565, 5 2016. DOI: 10.1145/2851581.2856492. URL https://dl.acm.org/doi/abs/10.1145/2851581.2856492?casa_token=gMv0Alf8b0MAAAAA:HHrVXCLJ2ngLrlws3fOeFLTSeWz9dGx-LlS1RA7q6P4TiCXIZ63SJKPYZMeN1HAHzEmNpmksNPThtg.

Y. Goldberg, O. Levy, T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean. word2vec Explained: deriving Mikolov et al.'s negative-sampling word-embedding method. 2 2014. URL http://arxiv.org/abs/1402.3722.

I. Goodfellow, Y. Bengio, and A. Courville. *Deep learning*. 2016. URL https://books.google.com/books?hl=en&lr=&id=omivDQAAQBAJ&oi=fnd&pg=PR5&dq=deep+learning&ots=MNO7bvpATU&sig=OSA4boYmvmB0pnTwuFr0F5ebJ8o.

B. Goodman and S. Flaxman. European Union regulations on algorithmic decision-making and a "right to explanation". *ojs.aaai.org*, 2017. URL https://ojs.aaai.org/index.php/aimagazine/article/view/2741.

S. Grigorescu, B. Trasnea, and T. Cocias. A survey of deep learning techniques for autonomous driving. *Wiley Online Library*. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/rob.21918?casa_token=eokqQF9GLLEAAAAA:isorsiFhjE_d5au2X9DErwIYc7ys-E-UZUIkFXAyfJntIVcvyKu9mTDjwz3OkS5dO3KiQpC1XW9Wc3NE.

J. Grudin. Three faces of human-computer interaction. *IEEE Annals of the History of Computing*, 27(4):46–62, 2005. ISSN 10586180. DOI: 10.1109/MAHC.2005.67.

J. Grudin. AI and HCI: Two fields divided by a common focus. *AI Magazine*, 30(4):48–57, 2009. ISSN 07384602. DOI: 10.1609/aimag.v30i4.2271. URL https://ojs.aaai.org/index.php/aimagazine/article/view/2271.

D. Gunning, M. Stefik, J. Choi, T. Miller, and S. Stumpf. XAI—Explainable artificial intelligence. *science.org*, 4(37):7120, 12 2019. DOI: 10.1126/scirobotics.aay7120. URL https://www.science.org/doi/abs/10.1126/scirobotics.aay7120.

C. Guo, G. Pleiss, Y. Sun, and K. Q. Weinberger. On calibration of modern neural networks. In *International Conference on Machine Learning*, pages 1321–1330. PMLR, 2017.

L. Guo, E. M. Daly, Alkan, M. Mattetti, O. Cornec, and B. Knijnenburg. Building Trust in Interactive Machine Learning via User Contributed Interpretable Rules. *dl.acm.org*, 22:12, 3 2022. DOI: 10.1145/3490099.3511111. URL https://dl.acm.org/doi/abs/10.1145/3490099.3511111?casa_token=z7vtIks4yXAAAAAA:hs7LVsOXasOgo31kBdYejsA_g7JlAFd9gHIiiQQs360tAkktSDrtUMD5vjOfkJJwN59aApodY-m0.

T. Hitron, Y. Orlev, I. Wald, A. Shamir, and H. Erel. Can children understand machine learning concepts? The effect of uncovering black boxes. *dl.acm.org*, page 11, 5 2019. DOI: 10.1145/3290605.3300645. URL https://dl.acm.org/doi/abs/10.1145/3290605.3300645?casa_token=rPy-ub207T0AAAAA:wKdgiWwKPS_I_fyVh55ixrL6Wg9nHvhlUkyRKt-dLXMQses-LsMj5pw-MR9iZnGr54G06Y_Xa7wdiQ.

S. S. Ho and H. Wechsler. Query by Transduction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(9):1557–1571, 9 2008. ISSN 01628828. DOI: 10.1109/TPAMI.2007.70811. URL http://ieeexplore.ieee.org/document/4384495/.

F. Hohman, K. Wongsuphasawat, M. B. Kery, and K. Patel. Understanding and Visualizing Data Iteration in Machine Learning. *Conference on Human Factors in Computing Systems - Proceedings*, 4 2020. DOI: 10.1145/3313831.3376177.

A. Holzinger. Interactive machine learning for health informatics: when do we need the human-in-the-loop? *Brain Informatics*, 3(2):119–131, 6 2016. ISSN 21984026. DOI: 10.1007/S40708-016-0042-6.

A. Holzinger and I. Jurisica. Knowledge discovery and data mining in biomedical informatics: The future is in integrative, interactive machine learning solutions. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 8401:1–18, 2014. ISSN 16113349. DOI: 10.1007/978-3-662-43968-5_1.

J. Hong, K. Lee, J. Xu, and H. Kacorri. Crowdsourcing the Perception of Machine Teaching. *dl.acm.org*, 4 2020. DOI: 10.1145/3313831.3376428. URL https://dl.acm.org/doi/abs/10.1145/3313831.3376428?casa_token=JIqlHhGMvE8AAAAA:d4GEJKNd_awd9n4OEyTEsS45AT6UbYXVa58V96UpP_cHxyEaPpM6o8d2BDtdmF45LmKpJDkzhoBLfA.

S. C. Hora. Aleatory and epistemic uncertainty in probability elicitation with an example from hazardous waste management. *Reliability Engineering & System Safety*, 54(2-3):217–223, 11 1996. ISSN 0951-8320. DOI: 10.1016/S0951-8320(96)00077-4.

E. Horvitz. Principles of mixed-initiative user interfaces. *Conference on Human Factors in Computing Systems - Proceedings*, pages 159–166, 1999. DOI: 10.1145/302979.303030. URL http://dx.doi.org/10.1145/2642918.2647408.

E. Horvitz. Reflections on challenges and promises of mixed-initiative interaction. *AI Magazine*, 28(2): 19–22, 2007. ISSN 07384602. URL https://ojs.aaai.org/index.php/aimagazine/article/view/2036.

A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam. MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. 4 2017. URL https://arxiv.org/abs/1704.04861v1.

K. Hsu, S. Levine, and C. Finn. Unsupervised learning via meta-learning. *7th International Conference on Learning Representations, ICLR 2019*, 2019.

E. Hüllermeier and W. Waegeman. Aleatoric and Epistemic Uncertainty in Machine Learning: An Introduction to Concepts and Methods.

M. I. Jordan and T. M. Mitchell. Machine learning: Trends, perspectives, and prospects. *Science*, 349(6245): 255–260, 2015.

D. M. Kane, S. Lovett, S. Moran, and J. Zhang. Active classification with comparison queries. *ieeexplore.ieee.org*, 2017. URL https://ieeexplore.ieee.org/abstract/document/8104072/.

S. Katan, M. Grierson, and R. Fiebrink. Using interactive machine learning to support interface development through workshops with disabled people. *dl.acm.org*, 2015-April:251–254, 4 2015. DOI: 10.1145/2702123.2702474. URL https://dl.acm.org/doi/abs/10.1145/2702123.2702474?casa_token=27hYvTTWifcAAAAA:8H54-sBjpUoC_cpAOAMfW3GtldVNtf-ylqZXUS4wMxxH3Q1tRJQUGljYW3-MzSwW8Nafzd86fM8iUA.

A. Kendall and Y. Gal. What Uncertainties Do We Need in Bayesian Deep Learning for Computer Vision? *Advances in Neural Information Processing Systems*, 30, 2017.

G. Kim and J. C. Ye. DiffusionCLIP: Text-Guided Diffusion Models for Robust Image Manipulation. 10 2021. URL http://arxiv.org/abs/2110.02711.

Y. Kim, K. Lee, and U. Oh. Understanding Interactive and Explainable Feedback for Supporting Non-Experts with Data Preparation for Building a Deep Learning Model. *International Journal of Advanced Smart Convergence*, 9(2):90–104, 2020. DOI: 10.7236/IJASC.2020.9.2.90. URL http://dx.doi.org/10.7236/IJASC.2020.9.2.90.

K. Koedinger, E. Brunskill, R. Baker, and E. McLaughlin. New potentials for data-driven intelligent tutoring system development and optimization. *ojs.aaai.org*, 2013. URL https://ojs.aaai.org/index.php/aimagazine/article/view/2484.

B. Koonce. MobileNetV3. *Convolutional Neural Networks with Swift for Tensorflow*, pages 125–144, 2021. DOI: 10.1007/978-1-4842-6168-2_11.

S. Krening. Newtonian Action Advice: Integrating Human Verbal Instruction with Reinforcement Learning. 4 2018. URL http://arxiv.org/abs/1804.05821.

S. Krening, B. Harrison, K. M. Feigh, C. L. Isbell, M. Riedl, and A. Thomaz. Learning from explanations using sentiment and advice in RL. *ieeexplore.ieee.org*, 9(1), 2017. DOI: 10.1109/TCDS.2016.2628365. URL https://ieeexplore.ieee.org/abstract/document/7742965/.

A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25:1097–1105, 2012.

T. Kulesza, S. Stumpf, M. Burnett, S. Yang, I. Kwan, and W. K. Wong. Too much, too little, or just right? Ways explanations impact end users' mental models. *Proceedings of IEEE Symposium on Visual Languages and Human-Centric Computing, VL/HCC*, pages 3–10, 2013. ISSN 19436092. DOI: 10.1109/VLHCC.2013.6645235.

T. Kulesza, S. Amershi, R. Caruana, and D. Fisher. Structured labeling for facilitating concept evolution in machine learning. *dl.acm.org*, pages 3075–3084, 2014. DOI: 10.1145/2556288.2557238.

T. Kulesza, M. Burnett, W. Wong, and S. Stumpf. Principles of explanatory debugging to personalize interactive machine learning. *dl.acm.org*, 2015-January:126–137, 3 2015. DOI: 10.1145/2678025.2701399. URL https://dl.acm.org/doi/abs/10.1145/2678025.2701399?casa_token=FoarFZnaEhAAAAAA:LSr-5b2o-yz-F5Ae2kNoGec9DIu4xoQksQxlwR8NuLeZIoXcc8Bry5g2geP3eXKzLU7tF5eI3jwdQg.

H. Lakkaraju, E. Kamar, R. Caruana, and E. Horvitz. Identifying unknown unknowns in the open world: Representations and policies for guided exploration. *aaai.org*. URL https://www.aaai.org/ocs/index.php/AAAI/AAAI17/paper/viewPaper/14434.

H. Lakkaraju, S. Bach, and J. Leskovec. Interpretable decision sets: A joint framework for description and prediction. *dl.acm.org*, 13-17-August-2016:1675–1684, 8 2016. DOI: 10.1145/2939672.2939874. URL https://dl.acm.org/doi/abs/10.1145/2939672.2939874?casa_token=vkY2JL0E7u0AAAAA:HuiGO10wonQBV667T-b32lwWpBAGh7UOc2ICZDALrcNwp3C8_oDxufsmITo-ekLnXSOsi2Tl2Qa9.

B. Lakshminarayanan, A. Pritzel, and C. Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. In *Advances in neural information processing systems*, pages 6402–6413, 2017.

A. B. L. Larsen, S. K. Sønderby, H. Larochelle, and O. Winther. Autoencoding beyond pixels using a learned similarity metric. *33rd International Conference on Machine Learning, ICML 2016*, 4:2341–2349, 2016.

Lecun, Y. The MNIST data of handwritten digits. *http://yann.lecun.com/exdb/mnist/*. URL https://ci.nii.ac.jp/naid/10027939599/.

J. Lee. A survey of robot learning from demonstrations for Human-Robot Collaboration. *arxiv.org*, 2017. URL http://arxiv.org/abs/1710.08789.

M. K. Lee, D. Kusbit, A. Kahng, J. T. Kim, X. Yuan, A. Chan, D. See, R. Noothigattu, S. Lee, A. Psomas, et al. Webuildai: Participatory framework for algorithmic governance. *Proceedings of the ACM on Human-Computer Interaction*, 3(CSCW):1–35, 2019.

D. D. Lewis, W. A. Gale, T. B. Laboratories, and M. Hill. A Sequential Algorithm for Training Text Classi ers 1 Introduction 3 An Uncertainty Sampling Algorithm 4 A Probabilistic Text Classi er. *Proceedings of the Seventeenth Annual International ACM_SIGIR conferrence on Research and Development in Information Retrieval*, pages 1–10, 1994.

D. Li, Y. Yang, Y. Song, and T. Hospedales. Learning to generalize: Meta-learning for domain generalization. *aaai.org*. URL https://www.aaai.org/ocs/index.php/AAAI/AAAI18/paper/viewFile/16067/16547.

Z. C. Lipton. The Mythos of Model Interpretability. *Queue*, 16(3):31–57, 6 2018. ISSN 1542-7730. DOI: 10.1145/3236386.3241340.

J. Liu, Z. Lin, S. Padhy, D. Tran, and T. Bedrax-Weiss. Simple and principled uncertainty estimation with deterministic deep learning via distance awareness. *arxiv.org*. URL https://arxiv.org/abs/2006.10108.

W. Liu, X. Wang, J. D. Owens, and Y. Li. Energy-based out-of-distribution detection. *Advances in Neural Information Processing Systems*, 2020-Decem, 2020. URL https://github.com/wetliu/.

R. Lomasky, C. E. Brodley, M. Aernecke, D. Walt, and M. Friedl. Active Class Selection. *Machine Learning: ECML 2007*, pages 640–647, 2007. DOI: 10.1007/978-3-540-74958-5_63.

M. Long, Y. Cao, J. Wang, M. I. Jordan, and J. Edu. Learning transferable features with deep adaptation networks. *proceedings.mlr.press*. URL http://proceedings.mlr.press/v37/long15.

C. C. Loy, T. M. Hospedales, T. Xiang, and S. Gong. Stream-based joint exploration-exploitation active learning. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 1560–1567, 2012.

W. Mackay. Using Video to Support Interaction Design. *DVD Tutorial, CHI*, 2002. URL http://www.cs.ubc.ca/~cs544/video/Mackay-using-video-usletter.pdf.

W. Mackay. Structured Observation to Support Interaction Design., 2014. URL https://scholar.google.com/scholar?hl=en&as_sdt=0%2C5&scioq=Structured+observation+to+support+interaction+design&q=Structured+Observation+to+Support+Interaction+Design&btnG=.

W. E. Mackay and A.-L. Fayard. HCI, natural science and design. pages 223–234, 1997. DOI: 10.1145/263552.263612.

J. Maiora, B. Ayerdi, and M. Graña. Random forest active learning for AAA thrombus segmentation in computed tomography angiography images. *Neurocomputing*, 126:71–77, 2014. ISSN 09252312. DOI: 10.1016/j.neucom.2013.01.051. URL http://dx.doi.org/10.1016/j.neucom.2013.01.051.

J. Markoff. Machines of loving grace: The quest for common ground between humans and robots. *ehubassist.anu.edu.au*. URL http://www.ehubassist.anu.edu.au/ag3k/18-dr-christophe-schumm/read-9780062266699-machines-of-loving-grace-the-quest-for-common-gr.pdf.

C. Meek. A Characterization of Prediction Errors. 11 2016. URL http://arxiv.org/abs/1611.05955.

M. L. Minsky and S. Papert. *Perceptrons, Expanded Edition An Introduction to Computational Geometry*. 1969. ISBN 9780262534772.

S. Mishra and J. M. Rzeszotarski. Designing interactive transfer learning tools for ml non-experts. *Conference on Human Factors in Computing Systems - Proceedings*, 5 2021. DOI: 10.1145/3411764.3445096. URL https://dl.acm.org/doi/abs/10.1145/3411764.3445096?casa_token=Q4EWopaQrPIAAAAA:b48tu4yocxBYfm8ikAxOeM0sGi_xVutaXPQuOiDGujZCWCtUH7tFadc1OBEjzkg8JIVEDXlamR_kyQ.

S. Mohamad, A. Bouchachia, and M. Sayed-Mouchaweh. A Bi-Criteria Active Learning Algorithm for Dynamic Data Streams. *IEEE Transactions on Neural Networks and Learning Systems*, 29(1):74–86, 2018. ISSN 21622388. DOI: 10.1109/TNNLS.2016.2614393.

D. Morris, R. F. P. computing, ubiquitous, and u. 2013. Using machine learning to support pedagogy in the arts. *Springer*, 2012. DOI: 10.1007/s00779-012-0526-1. URL https://link.springer.com/content/pdf/10.1007/s00779-012-0526-1.pdf.

K. Muhammad, A. Ullah, and J. Lloret. Deep learning for safe autonomous driving: Current challenges and future directions. *ieeexplore.ieee.org*. URL https://ieeexplore.ieee.org/abstract/document/9284628/.

J. Mukhoti, A. Kirsch, J. van Amersfoort, P. H. S. Torr, and Y. Gal. Deep Deterministic Uncertainty: A Simple Baseline. 2021. URL http://arxiv.org/abs/2102.11582.

I. Mukti and D. Biswas. Transfer learning based plant diseases detection using ResNet50. *ieeexplore.ieee.org*. URL https://ieeexplore.ieee.org/abstract/document/9068805/.

F. Ng, J. Suh, and G. Ramos. Understanding and supporting knowledge decomposition for machine teaching. *dl.acm.org*, pages 1183–1194, 7 2020. DOI: 10.1145/3357236.3395454. URL https://dl.acm.org/doi/abs/10.1145/3357236.3395454.

A. Nichol, P. Dhariwal, A. Ramesh, and P. Shyam. Glide: Towards photorealistic image generation and editing with text-guided diffusion models. *arxiv.org*. URL https://arxiv.org/abs/2112.10741.

A. Nichol, J. Achiam, and J. S. Openai. On First-Order Meta-Learning Algorithms. 3 2018. URL http://arxiv.org/abs/1803.02999.

S. Niu, Y. Liu, J. Wang, and H. Song. A Decade Survey of Transfer Learning (2010–2020). *IEEE Transactions on Artificial Intelligence*, 1(2):151–166, 6 2021. DOI: 10.1109/TAI.2021.3054609.

R. Northway, K. Hurley, C. O'connor, H. Thomas, S. Bale, A. Bevan, H. Board, A. House, and J. Howarth. Deciding what to research: an overview of a participatory workshop. *Wiley Online Library*, 42(4):323–327, 12 2014. DOI: 10.1111/bld.12080. URL https://onlinelibrary.wiley.com/doi/abs/10.1111/bld.12080?casa_token=VmPkhlaqKPcAAAAA:qgbpBo_ft3BhVdBxl7IfacbokP_4MNEjhPafrsiYuR0StJAgmVbP63X6bxgu5zi8sr92necaRawFbIKX.

K. Numa, K. Toriumi, K. Tanaka, M. Akaishi, and K. Hori. Participatory workshop as a creativity support system. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 5178 LNAI(PART 2):823–830, 2008. ISSN 16113349. DOI: 10.1007/978-3-540-85565-1_102.

C. Oh, S. Kim, J. Choi, J. Eun, S. Kim, J. Kim, J. Lee, and B. Suh. Understanding How People Reason about Aesthetic Evaluations of Artificial Intelligence. *dl.acm.org*, pages 1169–1181, 7 2020. DOI: 10.1145/3357236.3395430. URL https://dl.acm.org/doi/abs/10.1145/3357236.3395430.

M. Oquab, L. Bottou, and I. Laptev. Learning and transferring mid-level image representations using convolutional neural networks. *openaccess.thecvf.com*. URL http://openaccess.thecvf.com/content_cvpr_2014/html/Oquab_Learning_and_Transferring_2014_CVPR_paper.html.

S. J. Pan and Q. Yang. A Survey on Transfer Learning. URL http://socrates.acadiau.ca/courses/comp/dsilver/NIPS95.

A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper/2019/file/bdbca288fee7f92f2bfa9f7012727740-Paper.pdf.

M. E. Paté-Cornell. Uncertainties in risk analysis: Six levels of treatment. *Reliability Engineering and System Safety*, 54(2-3):95–111, 1996. ISSN 09518320. DOI: 10.1016/S0951-8320(96)00067-1.

K. R. Patil, X. Zhu, Kopeć, and B. C. Love. Optimal teaching for limited-capacity human learners. *Advances in Neural Information Processing Systems*, 3(January):2465–2473, 2014. ISSN 10495258. URL http://papers.nips.cc/paper/5541-bounded-regret-for-finite-armed-structured-bandits.pdf.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, et al. Scikit-learn: Machine learning in python. *the Journal of machine Learning research*, 12:2825–2830, 2011.

D. Pereira-Santos, R. B. C. Prudêncio, and A. C. C. de Carvalho. Empirical investigation of active learning strategies. *Neurocomputing*, 326-327:15–27. ISSN 18728286. DOI: 10.1016/j.neucom.2017.05.105. URL https://doi.org/10.1016/j.neucom.2017.05.105.

J. Postels, H. Blum, C. Cadena, R. Siegwart, L. van Gool, and F. Tombari. Quantifying aleatoric and epistemic uncertainty using density estimation in latent space. *arXiv*, 2020. ISSN 23318422. URL https://ui.adsabs.harvard.edu/abs/2020arXiv201203082P/abstract.

L. Pratt, J. Mostow, C. Kamm, and A. Kamm. Direct Transfer of Learned Information Among Neural Networks. *aaai.org*. URL https://www.aaai.org/Library/AAAI/1991/aaai91-091.php.

M. Racca, A. Oulasvirta, and V. Kyrki. Teacher-aware active robot learning. *ieeexplore.ieee.org*. URL https://ieeexplore.ieee.org/abstract/document/8673300/.

A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, G. Krueger, and I. Sutskever. Learning transferable visual models from natural language supervision. *proceedings.mlr.press*, 2021. URL http://proceedings.mlr.press/v139/radford21a.

I. Rahwan, M. Cebrian, N. Obradovich, J. Bongard, J.-F. Bonnefon, C. Breazeal, J. W. Crandall, N. A. Christakis, I. D. Couzin, M. O. Jackson, et al. Machine behaviour. *Nature*, 568(7753):477–486, 2019.

M. E. Ramirez-Loaiza, M. Sharma, G. Kumar, and M. Bilgic. Active learning: an empirical study of common baselines. *Data Mining and Knowledge Discovery*, 31(2):287–313, 2017. ISSN 1573756X. DOI: 10.1007/s10618-016-0469-7.

G. Ramos, C. Meek, P. Simard, J. Suh, and S. Ghorashi. Interactive machine teaching: a human-centered approach to building machine-learned models. *Taylor & Francis*, 35(5-6):413–451, 11 2020. DOI: 10.1080/07370024.2020.1734931. URL https://www.tandfonline.com/doi/abs/10.1080/07370024.2020.1734931?casa_token=lgbGPGFuuB0AAAAA:OyrWPC-TlsaOJ8oBasNYLQiAihJ6r-MBBJ7kM3m-1dI1OhY5xZZesa8OozcnDZMhCdtb4kMKpYx8rg.

H. Ravichandar, A. S. Polydoros, S. Chernova, and A. Billard. Recent Advances in Robot Learning from Demonstration. *Annual Review of Control, Robotics, and Autonomous Systems*, 3:297–330, 5 2020. ISSN 25735144. DOI: 10.1146/ANNUREV-CONTROL-100819-063206.

O. Razeghi, J. A. Solís-Lemus, A. W. Lee, R. Karim, C. Corrado, C. H. Roney, A. de Vecchi, and S. A. Niederer. CemrgApp: An interactive medical imaging application with image processing, computer vision, and machine learning toolkits for cardiovascular research. *SoftwareX*, 12, 7 2020. ISSN 23527110. DOI: 10.1016/J.SOFTX.2020.100570. URL https://doi.org/10.1016/j.softx.2020.100570.

M. Resnick, J. Maloney, A. Monroy-Hernández, N. Rusk, E. Eastmond, K. Brennan, A. Millner, E. Rosenbaum, J. Silver, B. Silverman, and Y. Kafai. Scratch: Programming for all. *Communications of the ACM*, 52 (11):60–67, 11 2009. ISSN 00010782. DOI: 10.1145/1592761.1592779.

M. T. Ribeiro, S. Singh, and C. Guestrin. "Why Should I Trust You?": Explaining the Predictions of Any Classifier. *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1135–1144, 2016. URL http://arxiv.org/abs/1602.04938.

G. Riccardi and D. Hakkani-Tür. Active learning: Theory and applications to automatic speech recognition. *IEEE Transactions on Speech and Audio Processing*, 13(4):504–510, 2005. ISSN 10636676. DOI: 10.1109/TSA.2005.848882.

K. Ridgeway and M. Mozer. Learning deep disentangled embeddings with the f-statistic loss. *proceedings.neurips.cc*. URL https://proceedings.neurips.cc/paper/2018/hash/2b24d495052a8ce66358eb576b8912c8-Abstract.html.

A. Rozantsev, M. Salzmann, and P. Fua. Beyond sharing weights for deep domain adaptation. *ieeexplore.ieee.org*. URL https://ieeexplore.ieee.org/abstract/document/8310033/.

C. Rudin. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature Machine Intelligence*, 1(5):206–215, 2019. ISSN 25225839. DOI: 10.1038/s42256-019-0048-x. URL https://www.nature.com/articles/s42256-019-0048-x.

M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L. C. Chen. MobileNetV2: Inverted Residuals and Linear Bottlenecks. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 4510–4520, 2018a. ISSN 10636919. DOI: 10.1109/CVPR.2018.00474. URL http://openaccess.thecvf.com/content_cvpr_2018/html/Sandler_MobileNetV2_Inverted_Residuals_CVPR_2018_paper.html.

M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4510–4520, 2018b.

N. Schnell, A. Röbel, D. Schwarz, G. Peeters, and R. Borghesi. MuBu and friends–assembling tools for content based real-time interactive audio processing in Max/MSP. *recherche.ircam.fr*. URL http://recherche.ircam.fr/anasyn/peeters/ARTICLES/Schnell_2009_ICMC_MUBU.pdf.

P. Schramowski, W. Stammer, S. Teso, A. Brugger, F. Herbert, X. Shao, H. G. Luigs, A. K. Mahlein, and K. Kersting. Making deep neural networks right for the right scientific reasons by interacting with their explanations. *Nature Machine Intelligence*, 2(8):476–486, 8 2020. ISSN 25225839. DOI: 10.1038/s42256-020-0212-3.

F. Schroff, D. Kalenichenko, and J. Philbin. Facenet: A unified embedding for face recognition and clustering. *cv-foundation.org*. URL https://www.cv-foundation.org/openaccess/content_cvpr_2015/html/Schroff_FaceNet_A_Unified_2015_CVPR_paper.html.

T. Scott, K. Ridgeway, and M. C. Mozer. Adapted deep embeddings: A synthesis of methods for k-shot inductive transfer learning. In *Advances in Neural Information Processing Systems*, pages 76–85, 2018.

D. Sculley, G. Holt, D. Golovin, E. Davydov, T. Phillips, D. Ebner, V. Chaudhary, and M. Young. Machine learning: The high interest credit card of technical debt. 2014. URL http://research.google/pubs/pub43146/.

H. Scurto and R. Fiebrink. *Grab-and-play mapping: Creative machine learning approaches for musical inclusion and exploration*. 2016. URL http://research.gold.ac.uk/id/eprint/18694/.

H. Scurto, B. Kerrebroeck, B. Caramiaux, and F. Bevilacqua. Designing deep reinforcement learning for human parameter exploration. *dl.acm.org*, pages 2013–2025, 6 2021. DOI: 10.1145/3461778.3462163. URL https://dl.acm.org/doi/abs/10.1145/3414472?casa_token=GVqxolqb8Q4AAAAA:yT5see3O89Es59I3-2R2gUOOPi9yKnptVJNUOZVQRBpU-GzLecz2-q4FGq_mXlabFZnBsT3-CRuNcQ.

B. Settles. Active Learning Literature Survey. Technical report, University of Wisconsin–Madison, 2010.

C. E. Shannon. A mathematical theory of communication. *The Bell system technical*, 1948. URL https://ieeexplore.ieee.org/abstract/document/6773024/.

D. Shen, G. Wu, and H. I. Suk. Deep Learning in Medical Image Analysis. *Annual Review of Biomedical Engineering*, 19:221–248, 6 2017. ISSN 15454274. DOI: 10.1146/ANNUREV-BIOENG-071516-044442.

A. Shinohara and S. Miyano. Teachability in computational learning. *New Generation Computing*, 8(4): 337–347, 12 1991. DOI: 10.1007/BF03037091.

P. Shivaswamy and T. Joachims. Online structured prediction via coactive learning. *Proceedings of the 29th International Conference on Machine Learning, ICML 2012*, 2:1431–1438, 2012.

P. Y. Simard, S. Amershi, D. M. Chickering, A. E. Pelton, S. Ghorashi, C. Meek, G. Ramos, J. Suh, J. Verwey, M. Wang, and J. Wernsing. Machine Teaching: A New Paradigm for Building Machine Learning Systems. *arxiv.org*, 2017. ISSN 2331-8422. URL https://arxiv.org/abs/1707.06742http://arxiv.org/abs/1707.06742.

D. Smilkov, N. Thorat, Y. Assogba, A. Yuan, N. Kreeger, P. Yu, K. Zhang, S. Cai, E. Nielsen, D. Soergel, S. Bileschi, M. Terry, C. Nicholson, S. N. Gupta, S. Sirajuddin, D. Sculley, R. Monga, G. Corrado, F. B. Viegas, and M. Wattenberg. Tensorflow.js: Machine learning for the web and beyond. 2019. URL https://arxiv.org/abs/1901.05350.

A. G. Smith, J. Petersen, C. Terrones-Campos, A. K. Berthelsen, N. J. Forbes, S. Darkner, L. Specht, and I. R. Vogelius. RootPainter3D: Interactive-machine-learning enables rapid and accurate contouring for radiotherapy. *Medical Physics*, 49(1):461–473, 1 2022. ISSN 00942405. DOI: 10.1002/MP.15353.

L. Smith and Y. Gal. Understanding measures of uncertainty for adversarial example detection. *34th Conference on Uncertainty in Artificial Intelligence 2018, UAI 2018*, 2:560–569, 2018. URL https://github.com/lsgos/uncertainty-adversarial-paper.

J. Snell, K. Swersky, and R. Zemel. Prototypical networks for few-shot learning. *proceedings.neurips.cc*. URL https://proceedings.neurips.cc/paper/6996-prototypical-networks-for-few-shot-learning.

D. Spiegelhalter and H. Riesch. Don't know, can't know: embracing deeper uncertainties when analysing risks. *royalsocietypublishing.org*, 369(1956):4730–4750, 12 2011. DOI: 10.1098/rsta.2011.0163. URL https://royalsocietypublishing.org/doi/abs/10.1098/rsta.2011.0163.

A. Strauss and J. Corbin. Basics of qualitative research techniques. 1998. URL https://resv.hums.ac.ir/uploads/22_288_57_1qualitative.pdf.

S. Stumpf, V. Rajaram, L. Li, W.-K. Wong, M. Burnett, T. Dietterich, E. Sullivan, and J. Herlocker. Interacting meaningfully with machine learning systems: Three experiments. *International Journal of Human-Computer Studies*, 67(8):639–662, 2009a.

S. Stumpf, V. Rajaram, L. Li, W.-K. Wong, M. Burnett, T. G. Dietterich, E. Sullivan, and J. Herlocker. Interacting meaningfully with machine learning systems: Three experiments. *Elsevier*, 67(8):639–662, 2009b. DOI: 10.1016/j.ijhcs.2009.03.004. URL https://www.sciencedirect.com/science/article/pii/S1071581909000457.

J. Suh, S. Ghorashi, G. Ramos, N.-C. Chen, S. Drucker, J. Verwey, P. Simard, M. Billinghurst, M. Burnett, A. Q. Authors', J. Suh, S. Ghorashi, G. Ramos, S. Drucker, J. Verwey, and P. Simard. AnchorViz: Facilitating Semantic Data Exploration and Concept Discovery for Interactive Ma-chine Learning. *ACM Transactions on Interactive Intelligent Systems*, 10(7), 2019. DOI: 10.1145/3241379. URL https://doi.org/10.1145/3241379.

N. Sultanum, S. Ghorashi, C. Meek, and G. Ramos. A teaching language for building object detection models. In *Proceedings of the 2020 ACM Designing Interactive Systems Conference*, pages 1223–1234, 2020.

C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013.

J. Talbot, B. Lee, A. Kapoor, and D. S. Tan. EnsembleMatrix: Interactive visualization to support machine learning with multiple classifiers. *Conference on Human Factors in Computing Systems - Proceedings*, pages 1283–1292, 2009. DOI: 10.1145/1518701.1518895.

C. Tan, F. Sun, T. Kong, W. Zhang, C. Yang, and C. Liu. A Survey on Deep Transfer Learning.

A. Thomaz and C. Breazeal. Teachable robots: Understanding human teaching behavior to build more effective robot learners. *Elsevier*. URL https://www.sciencedirect.com/science/article/pii/S000437020700135X.

A. Thomaz and G. Hoffman. Reinforcement learning with human teachers: Understanding how people want to teach robots. *ieeexplore.ieee.org*. URL https://ieeexplore.ieee.org/abstract/document/4107833/.

E. Ustinova and V. Lempitsky. Learning deep embeddings with histogram loss. *proceedings.neurips.cc*. URL https://proceedings.neurips.cc/paper/6464-learning-deep-embeddings-with-histogram-loss.

B. Ustun and C. Rudin. Optimized risk scores. *dl.acm.org*, Part F129685: 1125–1134, 8 2017. DOI: 10.1145/3097983.3098161. URL https://dl.acm.org/doi/abs/10.1145/3097983.3098161?casa_token=ts09Pz2-wH4AAAAA:FosYk9jawPLK9RUTpiJk0zexyqjenNM7LkoBdsDfmJUzxQK8n30MjBFBRB1p52VA7ZsP0tB025xA.

G. Valdes, J. Luna, E. Eaton, C. Simone, and L. Ungar. MediBoost: a patient stratification tool for interpretable decision making in the era of precision medicine. *nature.com*. URL https://www.nature.com/articles/srep37854.

J. Van Amersfoort, L. Smith, Y. W. Teh, and Y. Gal. Uncertainty Estimation Using a Single Deep Deterministic Neural Network. Technical report, 2020. URL https://github.com/y0ast/.

J. Vanschoren. Meta-Learning: A Survey. 10 2018. URL http://arxiv.org/abs/1810.03548.

R. Vilalta and Y. Drissi. A perspective view and survey of meta-learning. *Artificial Intelligence Review*, 18 (2):77–95, 10 2002. ISSN 02692821. DOI: 10.1023/A:1019956318069.

O. Vinyals, G. Deepmind, C. Blundell, T. Lillicrap, K. Kavukcuoglu, and D. Wierstra. Matching networks for one shot learning. *proceedings.neurips.cc*. URL https://proceedings.neurips.cc/paper/2016/hash/90e1357833654983612fb05e3ec9148c-Abstract.html.

D. Walker and F. Myrick. Grounded theory: An exploration of process and procedure. *Qualitative Health Research*, 16(4):547–559, 4 2006. ISSN 10497323. DOI: 10.1177/1049732305285972.

E. Wall, S. Ghorashi, and G. Ramos. Using Expert Patterns in Assisted Interactive Machine Learning: A Study in Machine Teaching. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 11748 LNCS:578–599, 2019. ISSN 16113349. DOI: 10.1007/978-3-030-29387-1_34.

B. C. Wallace, K. Small, C. E. Brodley, J. Lau, and T. A. Trikalinos. Deploying an Interactive Machine Learning System in an Evidence-Based Practice Center: abstrackr. 2012. URL http://github.com/bwallace/abstrackr-web.

J. Wang, F. Zhou, S. Wen, X. Liu, Y. Lin, and B. Research. Deep metric learning with angular loss. *openaccess.thecvf.com*. URL http://openaccess.thecvf.com/content_iccv_2017/html/Wang_Deep_Metric_Learning_ICCV_2017_paper.html.

Y. Wang, Q. Yao, J. T. Kwok, and L. M. Ni. Generalizing from a Few Examples: A Survey on Few-shot Learning. *ACM Computing Surveys*, 53(3), 6 2020. ISSN 15577341. DOI: 10.1145/3386252.

M. Ware, E. Frank, G. Holmes, M. Hall, and I. H. Witten. Interactive machine learning: Letting users build classifiers. *International Journal of Human Computer Studies*, 55(3):281–292, 9 2001. ISSN 10715819. DOI: 10.1006/ijhc.2001.0499. URL https://www.sciencedirect.com/science/article/abs/pii/S1071581901904999.

G. I. Webb, R. Hyde, H. Cao, H. L. Nguyen, F. Petitjean, G. I. Webb, R. Hyde, H. Cao, H. L. Nguyen, and F. Petitjean. Characterizing Concept Drift. 2016.

K. Weiss, T. M. Khoshgoftaar, and D. Wang Background. A survey of transfer learning. 2016. DOI: 10.1186/s40537-016-0043-6.

D. S. Weld and G. Bansal. The challenge of crafting intelligible intelligence. *Communications of the ACM*, 62 (6):70–79, 2019. ISSN 15577317. DOI: 10.1145/3282486.

C. T. Wolf, H. Zhu, J. Bullard, M. K. Lee, and J. R. Brubaker. The changing contours of "participation" in data-driven, algorithmic ecosystems: Challenges, tactics, and an agenda. *Proceedings of the ACM Conference on Computer Supported Cooperative Work, CSCW*, pages 377–384, 10 2018. DOI: 10.1145/3272973.3273005. URL https://doi.org/10.1145/3272973.3273005.

K. Wongsuphasawat, D. Smilkov, J. Wexler, J. Wilson, M. Mané, D. Fritz, D. Krishnan, F. B. Ví, and M. Wattenberg. Visualizing dataflow graphs of deep learning models in tensorflow. *ieeexplore.ieee.org*, 24(1), 2018. ISSN 1077-2626. DOI: 10.1109/TVCG.2017.2744878. URL https://ieeexplore.ieee.org/abstract/document/8019861/.

J. Xie, C. Myers, and J. Zhu. Interactive visualizer to facilitate game designers in understanding machine learning. *dl.acm.org*, 5 2019. DOI: 10.1145/3290607.3312851. URL https://dl.acm.org/doi/abs/10.1145/3290607.3312851?casa_token=S01iXy-rLi0AAAAA:e2oya9tQxosr71yUBouKCquSVK3zjWXj0jLG2i71R_496RVRGtdYC7tKd5dZU71XPFVKnyBP0Cga.

W. Xiong, L. Xie, S. Zhou, and J. Guan. Active learning for protein function prediction in protein-protein interaction networks. *Neurocomputing*, 145:44–52, 2014. ISSN 18728286. DOI: 10.1016/j.neucom.2014.05.075.

Y. Xu, H. Zhang, K. Miller, A. Singh, and A. Dubrawski. Noise-tolerant interactive learning using pairwise comparisons. *Advances in Neural Information Processing Systems*, 2017-Decem: 2432–2441, 2017. ISSN 10495258. URL https://proceedings.neurips.cc/paper/2017/hash/e11943a6031a0e6114ae69c257617980-Abstract.html.

Q. Yang, A. Scuito, J. Zimmerman, J. Forlizzi, and A. Steinfeld. Investigating how experienced UX designers effectively work with machine learning. *DIS 2018 - Proceedings of the 2018 Designing Interactive Systems Conference*, pages 585–596, 6 2018a. DOI: 10.1145/3196709.3196730.

Q. Yang, J. Suh, N. Chen, and G. Ramos. Grounding interactive machine learning tool design in how non-experts actually build models. *dl.acm.org*, pages 573–584, 6 2018b. DOI: 10.1145/3196709.3196729. URL https://dl.acm.org/doi/abs/10.1145/3196709.3196729?casa_token=-_FffPNcTYQAAAAA:ztCJL_KkqEKVh-uYK_8s4jNtERj5MZi7Ix2uPFs-lhdB1SrVRQd1ROAZlCNaxURtyAJkhv3pecqVEA.

J. Yosinski, J. Clune, Y. Bengio, and H. Lipson. How transferable are features in deep neural networks? In *Advances in neural information processing systems*, pages 3320–3328, 2014.

J. Yosinski, J. Clune, A. Nguyen, T. Fuchs, and H. Lipson. Understanding Neural Networks Through Deep Visualization. 6 2015. URL http://arxiv.org/abs/1506.06579.

M. Zeiler, D. Krishnan, and G. Taylor. Deconvolutional networks. *ieeexplore.ieee.org*. URL https://ieeexplore.ieee.org/abstract/document/5539957/.

Y. Zhang, Q. V. Liao, and R. K. Bellamy. Effect of confidence and explanation on accuracy and trust calibration in ai-assisted decision making. In *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*, pages 295–305, 2020.

Z. Zhang, G. Ning, and Z. He. Knowledge Projection for Deep Neural Networks. 10 2017. URL http://arxiv.org/abs/1710.09505.

J. Zhou, C. Bridon, F. Chen, A. Khawaji, and Y. Wang. Be informed and be involved: Effects of uncertainty and correlation on user's confidence in decision making. In *Proceedings of the 33rd Annual ACM Conference Extended Abstracts on Human Factors in Computing Systems*, pages 923–928, 2015.

X. Zhu. Machine teaching: An inverse problem to machine learning and an approach toward optimal education. *Proceedings of the National Conference on Artificial Intelligence*, 6:4083–4087, 2015. URL https://ojs.aaai.org/index.php/AAAI/article/view/9761.

X. Zhu, A. Singla, S. Zilles, and A. N. Rafferty. An Overview of Machine Teaching. 1 2018. URL http://arxiv.org/abs/1801.05927.

A. Zimmermann-Niefield, M. Turner, B. Murphy, S. K. Kane, and R. B. Shapiro. Youth learning machine learning through building models of athletic moves. *dl.acm.org*, pages 121–132, 6 2019a. DOI: 10.1145/3311927.3323139. URL https://dl.acm.org/doi/abs/10.1145/3311927.3323139?casa_token=XGMyTN7PqlkAAAAA:6HZt1JcjUmlp2LUeRJxOqDSTyWyCihmWbaycxqYb2VpQumKG1q08A0mg_k0pRiNS-EPE9mH9OKr05A.

A. Zimmermann-Niefield, M. Turner, B. Murphy, S. K. Kane, and R. B. Shapiro. Youth learning machine learning through building models of athletic moves. In *Proceedings of the 18th ACM International Conference on Interaction Design and Children*, pages 121–132, 2019b.

I. Zliobaite, A. Bifet, B. Pfahringer, and G. Holmes. Active learning with drifting streaming data. *IEEE Transactions on Neural Networks and Learning Systems*, 25(1):27–39, 2014. ISSN 2162237X. DOI: 10.1109/TNNLS.2012.2236570.