Deep Learning Uncertainty in Machine Teaching

Téo Sanchez, Baptiste Caramiaux, Pierre Thiel, Wendy E. Mackay

Deep Learning Uncertainty in Machine Teaching

TÉO SANCHEZ, Université Paris-Saclay, CNRS, Inria, LISN, France
BAPTISTE CARAMIAUX, Sorbonne Université, CNRS, ISIR, France
PIERRE THIEL, Sorbonne Université, CNRS, ISIR, France
WENDY E. MACKAY, Université Paris-Saclay, CNRS, Inria, LISN, France

Machine Learning models can output confident but incorrect predictions. To address this problem, ML researchers use various techniques to reliably estimate ML uncertainty, usually performed on controlled benchmarks once the model has been trained. We explore 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 setting. We are interested in users' perception of the difference between aleatoric and epistemic uncertainty and their use to teach and understand the classifier. We conducted an experiment where non-experts train a classifier to recognize card images, and are tested on their ability to predict classifier outcomes. Participants who used either larger or more varied training sets significantly improved their understanding of uncertainty, both epistemic or aleatoric. However, participants who relied on the uncertainty measure to guide their choice of training data did not significantly improve classifier training, nor were they better able to guess the classifier outcome. We identified three specific situations where participants successfully identified the difference between aleatoric and epistemic uncertainty: placing a card in the exact same position as a training card; placing different cards next to each other; and placing a non-card, such as their hand, next to or on top of a card. We discuss our methodology for estimating uncertainty for Interactive Machine Learning systems and question the need for two-level uncertainty in Machine Teaching.

CCS Concepts: • **Human-centered computing** → **Human computer interaction (HCI)**.

Additional Key Words and Phrases: ML uncertainty; Machine Teaching; Interactive Machine Learning; Human-AI Interaction; Human-centered analysis

ACM Reference Format:
Téo Sanchez, Baptiste Caramiaux, Pierre Thiel, and Wendy E. Mackay. 2022. Deep Learning Uncertainty in Machine Teaching. 1, 1 (February 2022), 26 pages. https://doi.org/10.1145/3490099.3511117

1 INTRODUCTION

Machine Learning (ML) has raised research interest that incorporates a human-computer interaction (HCI) perspective to increasingly involve end users, leading to the fields of Interactive Machine Learning (IML) and Human-AI Interaction [2, 10, 17]. One research area, Machine Teaching (MT), focuses on the human *teacher* of ML algorithms and their strategies for training ML models.

Increasing our understanding of this human perspective should improve the process by which domain experts convey relevant knowledge to a learning algorithm and, more generally, aid system developers in designing new types of interactive ML. However, we currently lack a clear understanding of how non-experts, users who are not trained in Computer Science and ML, interpret an ML algorithm's output as they teach new domain-specific concepts.

Authors' addresses: Téo Sanchez, Université Paris-Saclay, CNRS, Inria, LISN , Gif-sur-Yvette, France, teo.sanchez@inria.fr; Baptiste Caramiaux, Sorbonne Université, CNRS, ISIR, Paris, France, baptiste.caramiaux@sorbonne-universite.fr; Pierre Thiel, Sorbonne Université, CNRS, ISIR, Paris, France, pierre.thiel@sorbonne-universite.fr; Wendy E. Mackay, Université Paris-Saclay, CNRS, Inria, LISN, Gif-sur-Yvette, France, mackay@lri.fr.

Manuscript submitted to ACM

Although deep neural networks (DNN) have achieved state-of-the-art performance on image classification problems for over a decade [24], they are also prone to predicting false positives with high confidence levels [18]. Furthermore, barely perceptible input variations can easily deceive deep neural networks [45]. 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. 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.

Researchers have actively explored both aleatoric and epistemic uncertainty estimation in DNN on controlled, stereotyped data, such as Fashion MNIST [31]. 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 [4, 8, 50]. Confidence levels alone can be insufficient to improve AI-assisted decision making [50, 51]. Human-Computer Interaction (HCI) research has shown that the uncertainty inherent in probabilistic models can itself be considered as design material for interaction design [3]. To our knowledge, inspecting ML aleatoric and epistemic uncertainty has not been explored in the context of Interactive ML. 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. We investigate this assumption empirically through human-centered evaluations on a realistic teaching task with an IML system.

This article thus 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?

Through these questions, we hope to understand users' perception and use of aleatoric and epistemic uncertainty estimates in a machine teaching scenario. After reviewing the related research literature on uncertainty estimation and machine teaching, we 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. We use these results to select the appropriate method for an experiment investigating how non-experts understand both types of uncertainty. We use a teaching strategy designed for creative and educational domains [7, 35], where participants begin with an empty image classifier that makes random predictions and then trains it incrementally by selecting and presenting a series of images. We show that teaching decisions on training set size and data variability are more critical than the type of uncertainty participants were exposed to. We 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. We conclude with a discussion of how the results of the offline benchmark study and the experiment offer insights into the concepts of aleatoric ambiguity and epistemic novelty in the data from both the system and the human user's perspective.

2 BACKGROUND AND RELATED WORK

We present a background on uncertainty estimation in Deep Neural Networks. We then introduce Machine Teaching and the related work associated with this approach.

2.1 Background in Uncertainty Estimation in Deep Neural Networks

Before Machine Learning, the characterization of uncertainties and the manner of dealing with them was primarily the subjects of study of statisticians and engineers [12, 33, 44] and largely applied to risk analysis. These fields first introduced the distinction between epistemic and aleatoric [20], also called model and data uncertainty. The notion spread within the Machine Learning community much more recently, starting with the field of computer vision [23]. Uncertainty has been used with active learning, which aims to select the most informative instances to train the model and optimally reduce its epistemic uncertainty [38]. Active learning aims to train models with data that are costly to acquire. With the advent of Deep Learning and its adoption in many real-world applications, more effort has been made to develop methods capable of estimating the level and origin of uncertainty in a model prediction.

Aleatoric uncertainty captures the intrinsic randomness and ambiguity of the task. It is irreducible with further data. *Epistemic uncertainty* is caused by a lack of knowledge. It is reducible given additional information. 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. BNN emits predictions with uncertainty i.e. the errors margin in a data point prediction. Within the BNN approach, the distinction between aleatoric and epistemic uncertainty was first discussed by Kendall and Gal [23], showing neural network's limited awareness of its own confidence [22]. The research was driven by the necessity of determining if additional training data can resolve uncertainty. Theoretically, the mathematical formulation of uncertainty in BNN can be split in its reducible and irreducible contributions [43]. 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 [39]. In the following subsection, we present state-of-the-art techniques to estimate epistemic uncertainty.

Gal and colleagues [15] proposed a method to sample a trained model by randomly switching off a certain number of connections at inference (called *dropout*). Hence, one can derived 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 [25] 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 [16], or techniques involving end-to-end learning of uncertainty measures [9, 13] 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 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.

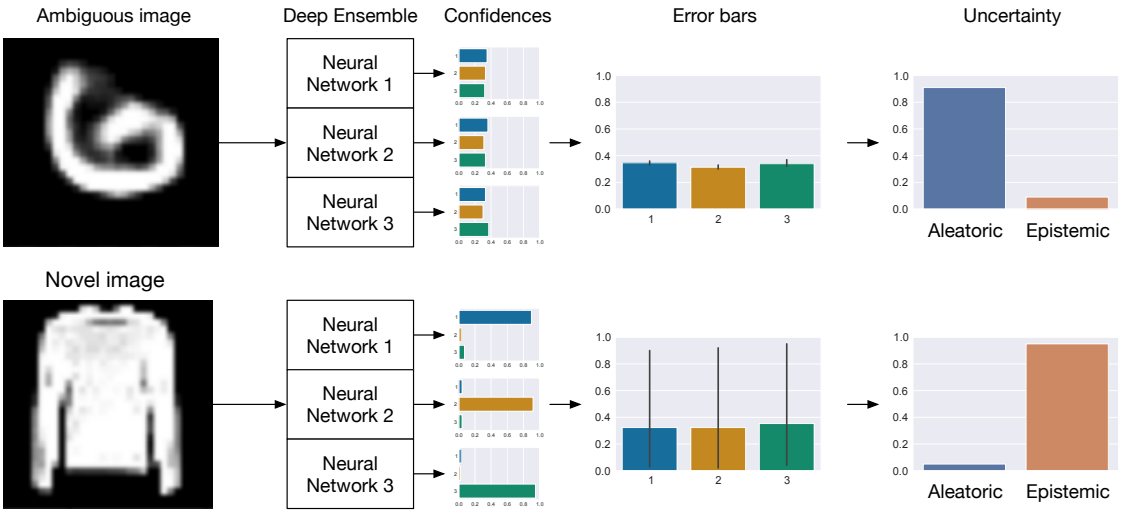


Fig. 1. Illustration of aleatoric and epistemic uncertainties through the Deep Ensemble approach, using as input data an ambiguous image with respect to the training set (made of handwritten digits recognition problem (MNIST)) and a novel image (unrelated to the training set).

Recently, uncertainty estimation has been tackled through a novel approach involving the use of feature space distances and density [27–29, 31, 47]. 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. Postels et al. [34] 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* [47], i.e. the fact that intermediate layers tend to map novel samples to the dense region of the feature space. Mukhoti et al. [31] 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. In the context of this work, we identify two main challenges. First, epistemic and aleatoric have not been evaluated within an IML workflow. Second, short iterations on model training is

crucial for the IML workflow. However, the computational cost of training DNN prevents incremental teaching from scratch. To enable incremental training from scratch, one can use pre-trained models for feature extraction along with a train a shallower network for classification [35]. This is a well-known technique in *transfer learning* [37] that was studied an IML context with novices [30]. As far as we know, the ML literature does not address the effect of transfer learning techniques on the DNN uncertainty estimations. We explore this problem in Section 3 and use the results to choose adequate uncertainty estimation for the user experiment in section 4.

2.2 Machine Teaching with novices

Machine Teaching (MT) originally refers to a mathematical problem that aims to find a minimal set of examples to train a target model [40, 41, 52]. In HCI, Machine Teaching refers to humans training ML models. Simard and colleagues [42] see MT as a way to improve the human teacher in the task of building machine learning models. The research in Machine Teaching is thus related to human-centered research in ML [17], but has also been investigated in Human-Robot Interaction (HRI) where a human trains a robot to learn given concepts [6, 46]. HCI-centred approach of MT primarily involves non-experts users as teachers (developers, domain-experts, or the general public). It aims to open up ML technology to a broader range of expertise.

Hong et al. [19] investigated how participants trained and deployed an image recognition application using images taken with their mobile phones. They focused their analysis on the type of variability induced in the data, showing that participants incorporate diversity in their examples, drawing parallels to how humans recognize objects independent of size, viewpoint, location, and illumination. Dwivedi et al. [11] conducted workshops with children in which they used MT to classify origami. They argued that MT could help children develop creativity and comfort with ML and AI. Their results proposed insights for designing MT experiences for children, including the fact that confidence scores and complementary metrics should be visible for experimentation and that the interface should allow quick data inspection. In a recent study, Sanchez et al. [35] investigated how adult novices teach an image classifier with sketches drawn by participants. They conducted a think-aloud study to reveal users' strategy and understanding while teaching the system. Among these results, they highlighted that optimization inertia of NN affects novices' understanding. They suggested guiding users in building a meaningful teaching curriculum i.e. a strategy that organizes the training examples to gradually introduce complexity.

In terms of guidance, Wall et al. [48] investigated how ML expert annotations and teaching strategies can serve to design notification guidance for novices in the task of training a classifier on articles. Although guidance did not improve the classifier's performances trained by novices, the task was found less difficult when novices had guidance. On the same line, Cakmak and Thomaz [6] studied how users spontaneously teach a binary classifier with examples from a finite set of combinations. They found that participants did not spontaneously generate optimal teaching sequences to improve mode accuracy, but this result can be leveraged with automated guidance.

In this work, teaching relied primarily on ML predictions, which can be misleading with complex tasks involving DNN. We explore the use of aleatoric and epistemic uncertainty as feedback for teaching DNNs, which is the focus of Section 4 and 5.

3 BENCHMARK STUDY

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 machine teaching experiment presented in the following sections.

3.1 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 machine teaching 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** [26] 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 [31]. We used 160 examples in the training set, 200 examples in the test set and 240 examples in the uncertain set.
(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 5). We collected the card images in the same lighting condition as the machine teaching 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.

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 [49]. To assess the impact of the feature extraction technique on uncertainty estimation, we consider three pre-trained models available online: *MobileNetV1* [21], *MobileNetV2* [36] and *ResNet50* [32].

3.2 Uncertainty estimation

3.2.1 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:

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

  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.

- 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.

3.2.2 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 [31].

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

$$H(z) = -\sum_{i=1}^{N} p_i(z) \log_2 p_i(z) \tag{2}$$

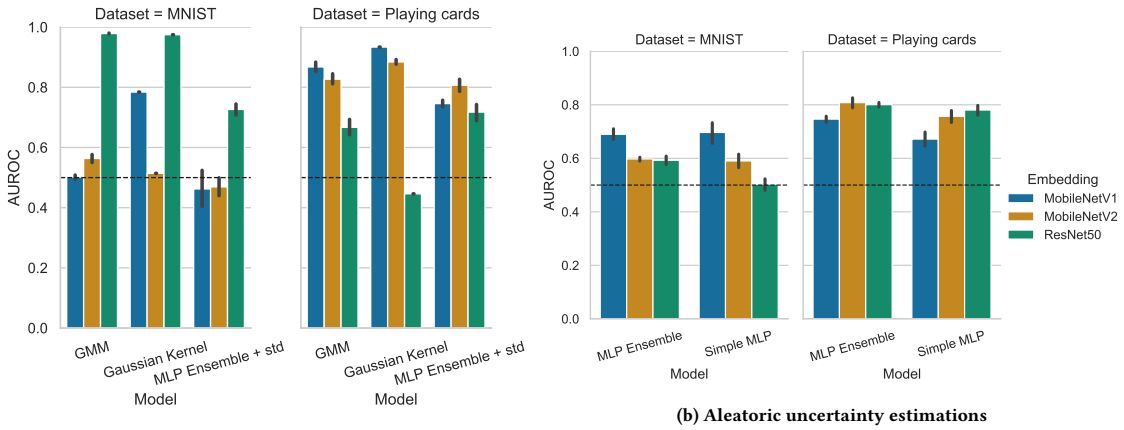
where z is an input data point and $p_i(z)$ is the softmax value for class i. We note that the uncertainty is calculated downstream from the predictions' probability emitted by the Neural Network.

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

3.3.1 Epistemic uncertainty estimation. Figure 2a 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.

3.3.2 Aleatoric uncertainty estimation. Figure 2b 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.



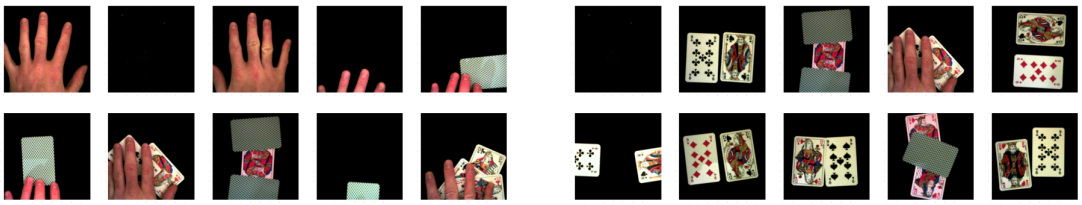
(a) Epistemic uncertainty estimation

(b) Aleatoric uncertainty estimations

Fig. 2. (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.

3.3.3 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 3 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.



(a) Epistemic uncertainty estimation

(b) Aleatoric uncertainty estimations

Fig. 4. 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.

To help the reader appreciate the data detected as uncertain, Figure 4 depicts the images located at the highest values of both uncertainty measures. We observed that high estimates of epistemic uncertainty showed out-of-distribution

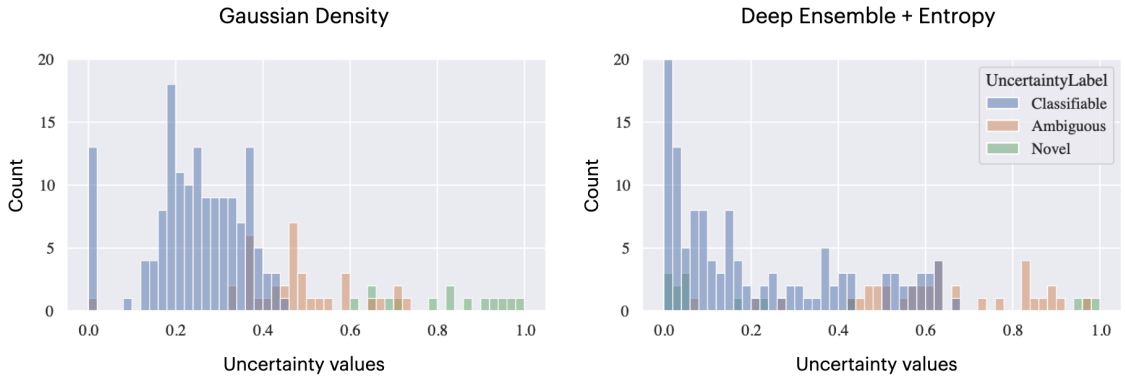


Fig. 3. Distribution of the playing card data according to (left) the epistemic uncertainty (Gaussian Kernel on MobileNetV1 features) and (right) the aleatoric uncertainty (entropy on MLP Ensemble using MobileNetV1 features). The label "classifiable" refers to data from the test set. Ambiguous and novel labels have been assigned to instances from the uncertain set by the first author.

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.

3.3.4 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 1 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 1. Entropy of the ten first components of the PCA on both datasets and the three different embeddings.

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.

4.1 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.

4.2 Setup

Apparatus: Figure 5:*(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 [1]. 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 5:*(bottom)* shows the experiment application, developed using the Marcelle [14] interactive machine learning (IML) toolkit for building interactive web interfaces based 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 4.3): *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 6 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 uncertainty is computed using the entropy on the MLPs' outputs. Epistemic uncertainty is computed through the Gaussian kernel density model.

4.3 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

[1]The source code of both the Marcelle application and the benchmark presented in section 3 is available at https://github.com/teo-sanchez/teaching-uncertainties-iui2022.

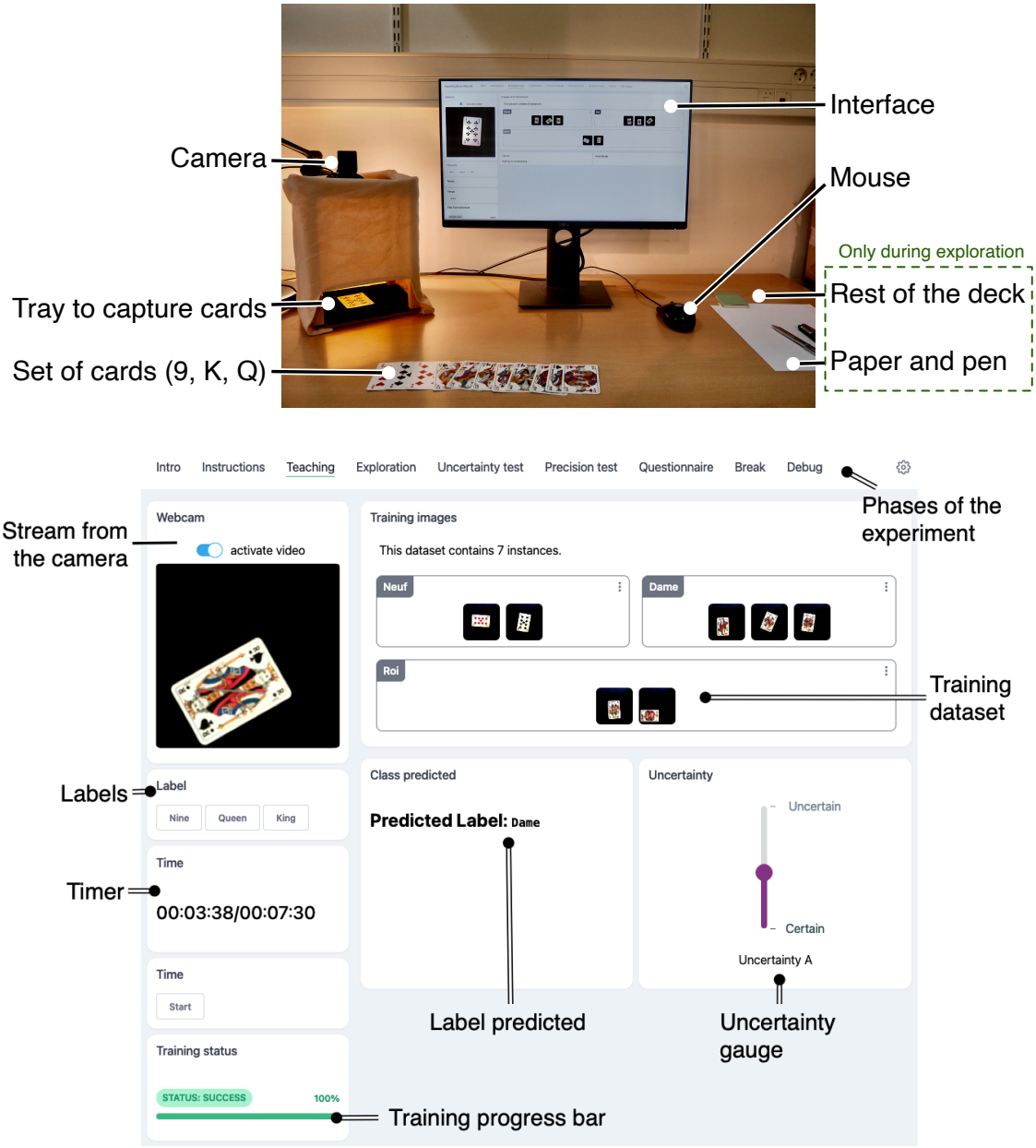


Fig. 5. ***Top:*** **The setup includes a screen, mouse, and camera stand for recording individual cards. Participants have access to the 12 training cards, the rest of the deck, paper and pens.** ***Bottom:*** **The application displays the live webcam feed, training set, prediction, uncertainty estimation 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.**

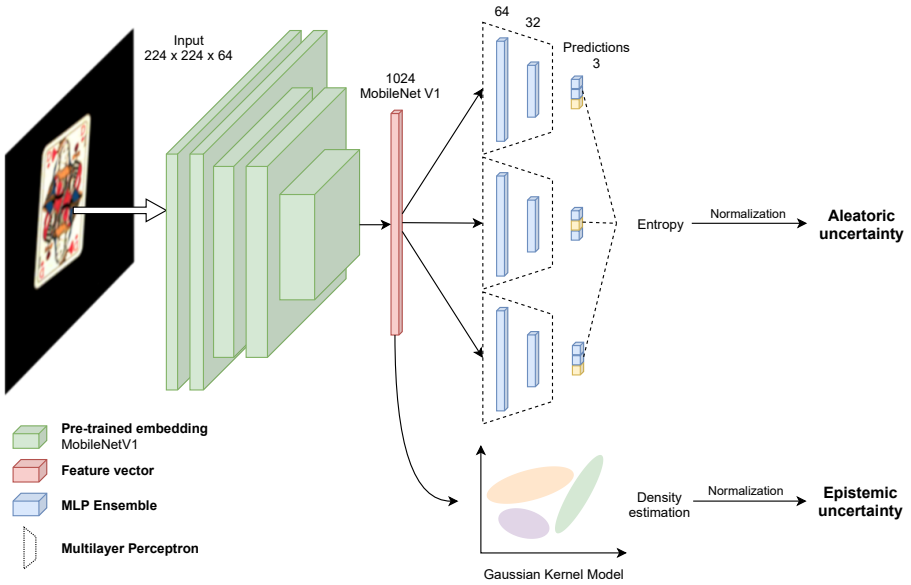


Fig. 6. Machine Learning pipeline and uncertainty estimation chosen for the user experiment. The first image represent a frame from the video stream. All the computation are performed in real-time. The prediction given on the interface is the averaged prediction over the MLP Ensemble.

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 teaching 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.

4.4 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 [5] 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:

$$score_{uncert} = 1 - \frac{1}{N}\sum_{i=1}^{N}|u_{model}(X) - u_{guessed}(X)| \tag{3}$$

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 4.3 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:

$$V_{training_set} = \frac{1}{3}\sum_{c\in classes}\frac{1}{C^2_{size(c)}}\sum_{X_i,X_j\in c} d(M(X_i), M(X_j)) \tag{4}$$

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 11 and Figure 12 in the Appendix depict the training sets of the most variable and least variable teaching sessions.

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 5.1, and studied through the quantitative analysis of the **uncertainty test** and **classification test** introduced in section 4.3. The results on (2) are presented in section 5.3 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).

5.1 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).

5.1.1 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 suggests 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 7. 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).

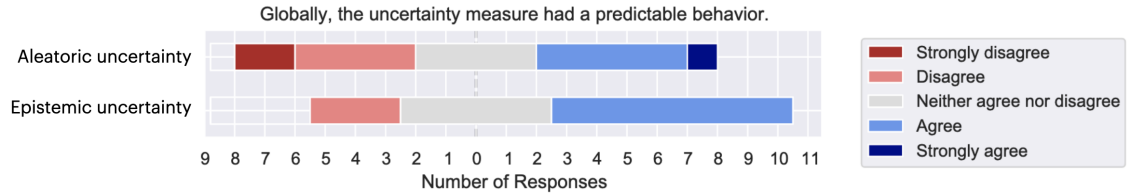


Fig. 7. Answers to the question "Globally, the uncertainty measure had a predictable behavior" exhibit a great variability across participants no matter the type of uncertainty shown as feedback.

5.1.2 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).

5.1.3 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.

5.1.4 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 8, as well the linear regressions between the size and variability of the training set and the participants' classification and uncertainty scores.

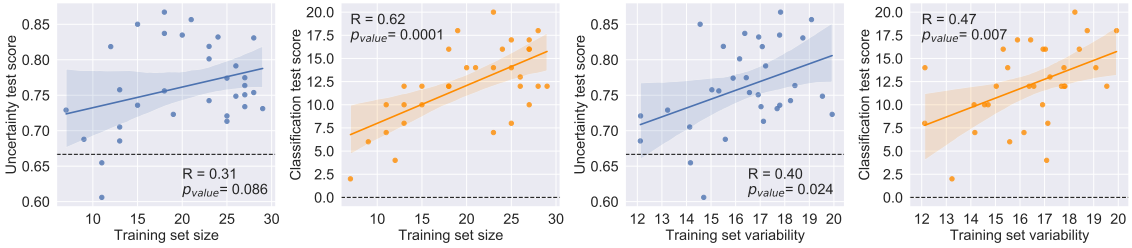


Fig. 8. Linear regressions between the training set size (the first two) and variability (the last two) and the participants' uncertainty test score (blue) and classification test score (orange). The dashed black lines represent the chance baseline i.e. random responses during the test phases.

The findings of this subsection 5.1 can be summarized as follow:

- 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.

5.2 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.

5.2.1 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 [38] where a model tries to select the most uncertain—therefore informative— instances in order to improve performance while reducing the amount of data resource.

5.2.2 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).

In summary:

- 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.

5.3 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 9.

Fig. 9. 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.

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.

5.3.1 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 9 which suggests that epistemic uncertainty is seen as more useful to identify images that the classifier does not know.

5.3.2 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).

5.3.3 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.

6 DISCUSSION

This section discusses the benchmark results and reflects on the use aleatoric and epistemic uncertainty in Interactive Machine Learning systems (IML). We then discuss the necessity and implications of distinguishing aleatoric from epistemic uncertainty.

6.1 Uncertainty in transfer learning: the importance of the 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 approach retrain the embedding using regularization techniques for ensuring sensitivity and smoothness of the feature space [31, 47]. In the context of IML, where iteration cycles are tight [2], 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. Rather than using out-of-the-box parameters from Imagenet, our MobileNetV1 architecture could have been retrained on several cards' decks with all values and different backgrounds and using feature-space regularization techniques. That said, the distribution of variance seems to be a promising tool to assess a pre-trained model in the context of uncertainty estimation and detection. We encourage further research to understand Transfer Learning for uncertainty estimation with this approach.

6.2 On the difficulty of providing annotations of uncertain instances

As mentioned in the related work in Section 2, aleatoric uncertainty is commonly associated with ambiguity and noise in the data. Our benchmark study and our results showed that two different cards on the same image led to high ambiguity (i.e. high aleatoric uncertainty). It suggests that two cards on an image remains a *concept* close to what it has been learned. By contrast, epistemic uncertainty is usually associated with novelty with respect to the training examples. Our results showed that a hand in the image leads to high novelty (i.e. high epistemic uncertainty), suggesting that a hand is seen as a different *concept* by the classifier. Researchers usually rely on annotated data that distinguish ambiguous from novel instances to evaluate both uncertainty estimates. However, it might be very challenging for users 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. [31] 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.

6.3 On using uncertainty as a guide

We found that participants perceived differences between aleatoric and epistemic uncertainty when exploring extreme situations: a card in the exact same position as a training example, two different cards next to each other, or placing an unrelated object (e.g. a hand) within the frame. This finding suggests that it would be beneficial to differentiate when users need to (1) explore the tail of the uncertainty distribution (extreme values) or (2) teach the classifier and foresee its outcomes. The first case applies to domain experts who explicitly want to understand the examples on which a model may be uncertain. For instance, a medical doctor may have a large but noisy dataset. Clinicians could benefit from an IML system showing both ambiguity or novelty estimates for some chosen input examples to assist decision-making. Clinicians could then understand the intrinsic uncertainty of the machine in their data and browse novel data to select the ones that the classifier should take into account. Thus, we foresee a promising research direction in how IML could enable users to discover edge-cases examples with high uncertainty values. Combining data augmentation techniques with meaningful interactions could help users discover such instances.

6.4 On the (absence of perceived) difference between aleatoric and epistemic uncertainties

Participants using uncertainty as a guide did not train a more accurate model, nor were they more able to correctly foresee their classifier 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 were better at predicting their classifier's outcomes regarding both classification and uncertainty estimation. We presume that participants opting for a structured curriculum might have a better mental picture of their training set content. Hence, encouraging structured rather than uncertainty-based curricula would be indicated if we want to improve users' general understanding of their classifier. These results also 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. It might instead be helpful when refining the model.

We assume that the perception of ML uncertainties may change when scaling with larger and more complex data (more variable inputs and more classes). In this case, we suspect that users might lose track of the content of their dataset, and consequently, why a specific example is uncertain. Therefore, we believe that future work on machine learning uncertainty should be done in close collaboration with research on explainable AI, and explore how data complexity affect their ability to understand and track their network's uncertainty. A promising approach would be to investigate why a piece of data is uncertain, just as Explainable AI focuses on explaining the model's predictions.

Finally, ML research suggests that the type of uncertainty can be seen as a continuum from aleatoric to epistemic, captured across the network's layers [34]. Epistemic uncertainty is captured in shallower layers of a DNN, whereas aleatoric uncertainty is captured in deeper layers. Considering the relation between continuous uncertainty estimate and model architecture is a promising research direction that could lead to novel uncertainty visualization and interaction techniques.

7 CONCLUSION

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 classifier. 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. This result suggests that the distinction between aleatoric and epistemic uncertainty can be made relevant for domain experts, e.g. medical practitioners, who specifically explore the extreme values in the "tail" of an uncertainty distribution. However, it also suggests that two-level uncertainty may

not be relevant for performance-oriented Machine Teaching or when users seek a more general understanding of the classifier's strengths and weaknesses.

Our results bring a human-centered perspective to a theoretical and computational problem—uncertainty estimation in neural networks—that may be beneficial to several fields including Explainable AI, Interactive Machine Learning and Human-centered AI. We hope that this work will encourage other researchers to apply the concepts of aleatoric and epistemic uncertainty to the design of more usable and transparent ML tools accessible to a broader range of users.

ACKNOWLEDGMENTS

This research was supported by the ARCOL project (ANR-19-CE33-0001) – Interactive Reinforcement Co-Learning, from the French National Research Agency. We want to acknowledge and thank everyone involved in each stage of the research. We want to express our sincere gratitude to the anonymous reviewers and the participants in the pilot study and the experiment. Thanks to Gianni Franchi for his valuable thoughts on uncertainty in Deep Neural Networks.

REFERENCES

[1] [n. d.]. French-suited playing cards with the Paris pattern. https://en.wikipedia.org/wiki/French-suited_playing_cards#Paris_pattern

[2] Saleema Amershi, Maya Cakmak, W. Bradley Knox, and Todd Kulesza. 2014. Power to the people: The role of humans in interactive machine learning. *AI Magazine* 35, 4 (2014), 105–120. https://doi.org/10.1609/aimag.v35i4.2513

[3] Jesse Josua Benjamin, Arne Berger, Nick Merrill, and James Pierce. 2021. Machine Learning Uncertainty as a Design Material: A Post-Phenomenological Inquiry. (2021). https://doi.org/10.1145/3411764.3445481

[4] Umang Bhatt, Javier Antorán, Yunfeng Zhang, Q Vera Liao, Prasanna Sattigeri, Riccardo Fogliato, Gabrielle Melançon, Ranganath Krishnan, Jason Stanley, Omesh Tickoo, et al. 2021. Uncertainty as a form of transparency: Measuring, communicating, and using uncertainty. In *Proceedings of the 2021 AAAI/ACM Conference on AI, Ethics, and Society*. 401–413.

[5] Virginia Braun and Victoria Clarke. 2006. Using thematic analysis in psychology. *Qualitative research in psychology* 3, 2 (2006), 77–101.

[6] Maya Cakmak and Andrea L. Thomaz. 2014. Eliciting good teaching from humans for machine learners. *Artificial Intelligence* 217 (12 2014), 198–215. https://doi.org/10.1016/j.artint.2014.08.005

[7] Michelle Carney, Barron Webster, Irene Alvarado, Kyle Phillips, Noura Howell, Jordan Griffith, Jonas Jongejan, Amit Pitaru, and Alexander Chen. 2020. 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*. 1–8.

[8] Eoin Delaney, Derek Greene, and Mark T Keane. 2021. Uncertainty Estimation and Out-of-Distribution Detection for Counterfactual Explanations: Pitfalls and Solutions. *arXiv preprint arXiv:2107.09734* (2021).

[9] Terrance DeVries and Graham W Taylor. 2018. Learning confidence for out-of-distribution detection in neural networks. In *arXiv preprint arXiv:1802.04865*.

[10] John J Dudley and Per Ola Kristensson. 2018. A review of user interface design for interactive machine learning. *ACM Transactions on Interactive Intelligent Systems (TiiS)* 8, 2 (2018), 1–37.

[11] Utkarsh Dwivedi, Jaina Gandhi, Raj Parikh, Merijke Coenraad, Elizabeth Bonsignore, and Hernisa Kacorri. [n. d.]. Exploring Machine Teaching with Children. ([n. d.]).

[12] MH Faber. 2005. On the treatment of uncertainties and probabilities in engineering decision analysis. (2005). https://asmedigitalcollection.asme.org/offshoremechanics/article-abstract/127/3/243/468180

[13] Gianni Franchi, Andrei Bursuc, Emanuel Aldea, Séverine Dubuisson, and Isabelle Bloch. 2020. *TRADI: Tracking deep neural network weight distributions*. Technical Report. https://hal.archives-ouvertes.fr/hal-02922336

[14] Jules Françoise, Baptiste Caramiaux, and Téo Sanchez. 2021. Marcelle: Composing Interactive Machine Learning Workflows and Interfaces;. (2021). https://doi.org/10.1145/3472749.3474734{i}

[15] Yarin Gal. 2016. Uncertainty in deep learning. *University of Cambridge* 1 (2016), 3.

[16] Yarin Gal and Zoubin Ghahramani. 2015. Dropout as a Bayesian Approximation: Appendix. (6 2015). http://arxiv.org/abs/1506.02157

[17] Marco Gillies, Rebecca Fiebrink, Atau Tanaka, Jérémie Garcia, Frédéric Bevilacqua, Alexis Heloir, Fabrizio Nunnari, Wendy Mackay, Saleema Amershi, Bongshin Lee, and others. 2016. Human-centred machine learning. In *Proceedings of the 2016 CHI Conference Extended Abstracts on Human Factors in Computing Systems*. 3558–3565.

[18] Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. 2017. On calibration of modern neural networks. In *International Conference on Machine Learning*. PMLR, 1321–1330.

[19] Jonggi Hong, Kyungjun Lee, June Xu, and Hernisa Kacorri. 2020. Crowdsourcing the Perception of Machine Teaching. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*. ACM, New York, NY, USA, 1–14. https://doi.org/10.1145/3313831.3376428

[20] Stephen C. Hora. 1996. Aleatory and epistemic uncertainty in probability elicitation with an example from hazardous waste management. *Reliability Engineering & System Safety* 54, 2-3 (11 1996), 217–223. https://doi.org/10.1016/S0951-8320(96)00077-4

[21] Andrew G. Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. 2017. MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. (4 2017). https://arxiv.org/abs/1704.04861v1

[22] Eyke Hüllermeier and Willem Waegeman. [n. d.]. Aleatoric and Epistemic Uncertainty in Machine Learning: An Introduction to Concepts and Methods. ([n. d.]).

[23] Alex Kendall and Yarin Gal. 2017. What Uncertainties Do We Need in Bayesian Deep Learning for Computer Vision? *Advances in Neural Information Processing Systems* 30 (2017).

[24] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. 2012. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems* 25 (2012), 1097–1105.

[25] Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. 2017. Simple and scalable predictive uncertainty estimation using deep ensembles. In *Advances in neural information processing systems*. 6402–6413.

[26] LECUN and Y. [n. d.]. THE MNIST DATABASE of handwritten digits. *http://yann.lecun.com/exdb/mnist/* ([n. d.]). https://ci.nii.ac.jp/naid/10027939599/

[27] Min Kyung Lee, Daniel Kusbit, Anson Kahng, Ji Tae Kim, Xinran Yuan, Allissa Chan, Daniel See, Ritesh Noothigattu, Siheon Lee, Alexandros Psomas, et al. 2019. WeBuildAI: Participatory framework for algorithmic governance. *Proceedings of the ACM on Human-Computer Interaction* 3, CSCW (2019), 1–35.

[28] JZ Liu, Z Lin, S Padhy, D Tran, T Bedrax-Weiss arXiv preprint arXiv …, and undefined 2020. [n. d.]. Simple and principled uncertainty estimation with deterministic deep learning via distance awareness. *arxiv.org* ([n. d.]). https://arxiv.org/abs/2006.10108

[29] Weitang Liu, Xiaoyun Wang, John D. Owens, and Yixuan Li. 2020. Energy-based out-of-distribution detection. *Advances in Neural Information Processing Systems* 2020-December (2020). https://github.com/wetliu/

[30] S Mishra, JM Rzeszotarski Proceedings of the 2021 CHI Conference, and undefined 2021. 2021. Designing Interactive Transfer Learning Tools for ML Non-Experts. *dl.acm.org* (5 2021). https://doi.org/10.1145/3411764.3445096

[31] Jishnu Mukhoti, Andreas Kirsch, Joost van Amersfoort, Philip H. S. Torr, and Yarin Gal. 2021. Deterministic Neural Networks with Appropriate Inductive Biases Capture Epistemic and Aleatoric Uncertainty. (2021). http://arxiv.org/abs/2102.11582

[32] IZ Mukti, D Biswas 2019 4th International Conference on, and undefined 2019. [n. d.]. Transfer learning based plant diseases detection using ResNet50. *ieeexplore.ieee.org* ([n. d.]). https://ieeexplore.ieee.org/abstract/document/9068805/

[33] M. Elisabeth Paté-Cornell. 1996. Uncertainties in risk analysis: Six levels of treatment. *Reliability Engineering and System Safety* 54, 2-3 (1996), 95–111. https://doi.org/10.1016/S0951-8320(96)00067-1

[34] Janis Postels, Hermann Blum, Cesar Cadena, Roland Siegwart, Luc van Gool, and Federico Tombari. 2020. Quantifying aleatoric and epistemic uncertainty using density estimation in latent space. *arXiv* (2020). https://ui.adsabs.harvard.edu/abs/2020arXiv201203082P/abstract

[35] Téo Sanchez, Baptiste Caramiaux, Jules Françoise, Frédéric Bevilacqua, and Wendy E. Mackay. 2021. How do People Train a Machine? Strategies and (Mis)Understandings. *Proceedings of the ACM on Human-Computer Interaction* 5, CSCW1 (4 2021), 1–26. https://doi.org/10.1145/3449236

[36] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. 2018. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 4510–4520.

[37] Tyler Scott, Karl Ridgeway, and Michael C Mozer. 2018. Adapted deep embeddings: A synthesis of methods for k-shot inductive transfer learning. In *Advances in Neural Information Processing Systems*. 76–85.

[38] Burr Settles. 2009. Active Learning Literature Survey - TR 1648. *Sciences-New York* (2009). https://doi.org/10.1016/j.matlet.2010.11.072

[39] Claude E. Shannon. 1948. A mathematical theory of communication. *The Bell system technical* (1948). https://ieeexplore.ieee.org/abstract/document/6773024/

[40] Ayuni Shinohara and Satoru Miyano. 1989. A Foundation of Algorithmic Teaching. *Information Systems* 092, 541 (1989). https://catalog.lib.kyushu-u.ac.jp/ja/recordID/3129/?repository=yes

[41] Ayumi Shinohara and Satoru Miyano. 1991. Teachability in computational learning. *New Generation Computing* 8, 4 (12 1991), 337–347. https://doi.org/10.1007/BF03037091

[42] Patrice Y. Simard, Saleema Amershi, David M. Chickering, Alicia Edelman Pelton, Soroush Ghorashi, Christopher Meek, Gonzalo Ramos, Jina Suh, Johan Verwey, Mo Wang, and John Wernsing. 2017. Machine teaching a new paradigm for building machine learning systems. *arXiv* (2017).

[43] Lewis Smith and Yarin Gal. 2018. Understanding measures of uncertainty for adversarial example detection. *34th Conference on Uncertainty in Artificial Intelligence 2018, UAI 2018* 2 (2018), 560–569. https://github.com/lsgos/uncertainty-adversarial-paper

[44] DJ Spiegelhalter, H Riesch Transactions of the Royal …, and undefined 2011. 2011. Don't know, can't know: embracing deeper uncertainties when analysing risks. *royalsocietypublishing.org* 369, 1956 (12 2011), 4730–4750. https://doi.org/10.1098/rsta.2011.0163

[45] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. 2013. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199* (2013).

[46] Andrea L. Thomaz and Maya Cakmak. 2009. Learning about objects with human teachers. *Proceedings of the 4th ACM/IEEE international conference on Human robot interaction* (2009), 15. https://doi.org/10.1145/1514095.1514101

[47] Joost Van Amersfoort, Lewis Smith, Yee Whye Teh, and Yarin Gal. 2020. *Uncertainty Estimation Using a Single Deep Deterministic Neural Network*. Technical Report. https://github.com/y0ast/

[48] Emily Wall, Soroush Ghorashi, and Gonzalo Ramos. 2019. 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 (2019), 578–599. https://doi.org/10.1007/978-3-030-29387-1{_}34

[49] Yaqing Wang, Quanming Yao, James T. Kwok, and Lionel M. Ni. 2020. Generalizing from a Few Examples: A Survey on Few-shot Learning. *Comput. Surveys* 53, 3 (6 2020). https://doi.org/10.1145/3386252

[50] Yunfeng Zhang, Q Vera Liao, and Rachel KE Bellamy. 2020. 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*. 295–305.

[51] Jianlong Zhou, Constant Bridon, Fang Chen, Ahmad Khawaji, and Yang Wang. 2015. 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*. 923–928.

[52] Xiaojin Zhu. 2015. Machine Teaching: An Inverse Problem to Machine Learning and an Approach Toward Optimal Education. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligenc*. 4083–4087.

APPENDIX A

The Deep Ensemble and feature-based approach

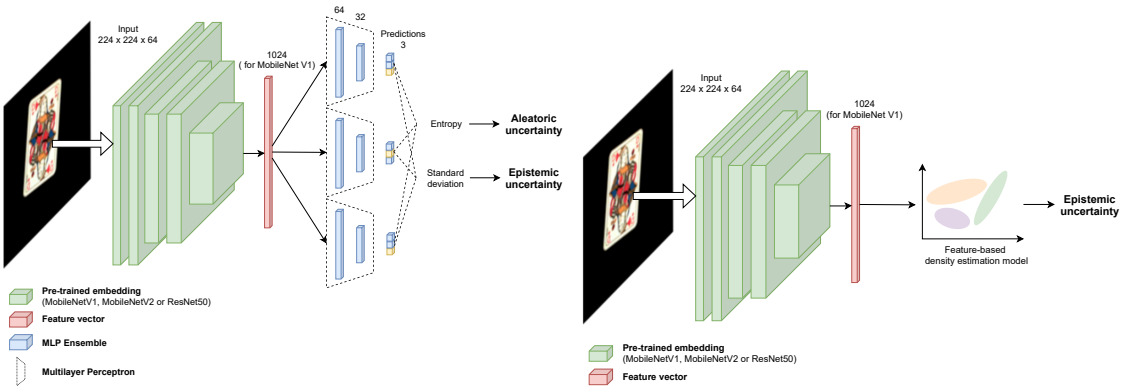


Fig. 10. Schema of the Deep Ensemble (Left) and the feature-based (Right) approaches.

Summary table of the benchmark conducted

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

APPENDIX B

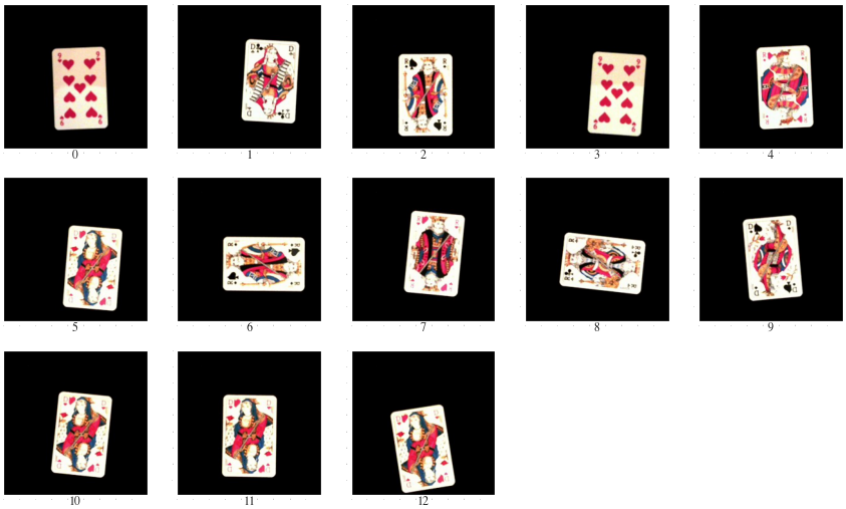


Fig. 11. The least variable training set (participant 15) from the teaching sessions of the participants.

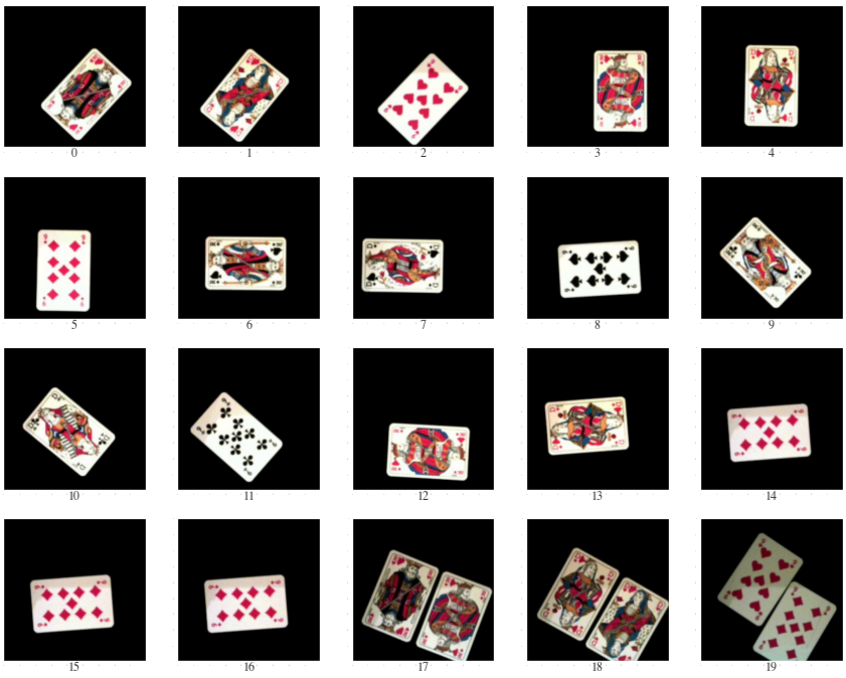


Fig. 12. The most variable training set (participant 4) from the teaching sessions of the participants.