Designing Interactive Transfer Learning Tools for ML Non-Experts

Swati Mishra
Cornell University
swati@infosci.cornell.edu

Jeffrey M Rzeszotarski
Cornell University
jeffrz@cornell.edu

ABSTRACT

Interactive machine learning (iML) tools help to make ML accessible to users with limited ML expertise. However, gathering necessary training data and expertise for model-building remains challenging. Transfer learning, a process where learned representations from a model trained on potentially terabytes of data can be transferred to a new, related task, offers the possibility of providing "building blocks" for non-expert users to quickly and effectively apply ML in their work. However, transfer learning largely remains an expert tool due to its high complexity. In this paper, we design a prototype to understand non-expert user behavior in an interactive environment that supports transfer learning. Our findings reveal a series of data- and perception-driven decision-making strategies non-expert users employ, to (in)effectively transfer elements using their domain expertise. Finally, we synthesize design implications which might inform future interactive transfer learning environments.

CCS CONCEPTS

• **Human-centered computing** → **User models**; **Graphical user interfaces**; *User centered design.*

KEYWORDS

Interactive Machine Learning, User Study, Transfer Learning, Prototyping

ACM Reference Format:
Swati Mishra and Jeffrey M Rzeszotarski. 2021. Designing Interactive Transfer Learning Tools for ML Non-Experts. In *CHI Conference on Human Factors in Computing Systems (CHI '21), May 8–13, 2021, Yokohama, Japan.* ACM, New York, NY, USA, 15 pages. https://doi.org/10.1145/3411764.3445096

1 INTRODUCTION

Machine Learning (ML) systems have seen adoption in fields outside of computer science such as healthcare, finance, manufacturing and even marketing [27]. This has powered a growing body of research in interactive machine learning (iML) [16], integrating knowledge from both human-computer interaction (HCI) and ML [29, 41]) to make these systems accessible to diverse audiences. While stakeholders outside of computing domains may lack formal ML expertise/training, they may possess domain expertise in other areas which can significantly benefit from the use of ML systems.

Well-designed iML tools have proven to be helpful in supporting ML non-experts to integrate their expertise in a traditional model development environment. Specific projects have investigated iML when designing models [52], selecting features [31, 50], and evaluating results [17]. However, comparably little research has examined the inverse – *how iML tools might be designed to assist non-experts in efficiently integrating ML into their own working environment.* For instance, a mobile application designer might want to employ a classification model to identify sensitive user generated content in their application prior to publishing it to the web. However, in their case, learning about ML and making sense of the ML model development workflow may be too onerous despite the potential benefits ML might bring to their real-world problems [57]. In this work, we consider transfer learning as a potential technique to encourage novice appropriation of existing, performant models as "building blocks". Instead of promoting end-to-end interactive model development, we explore interactive environments that help non-experts build models by re-purposing the components of existing, expert curated models through transfer learning.

Transfer Learning [51] is the technique of adapting a model trained on one task to another, related task, through the transfer of relevant model features. It is widely used in the ML community to push the limits of model performance, and has been instrumental in the growth of computer vision [34, 38] and natural language processing [45]. Further, the ML community also promotes transfer learning across varied applications by creating repositories of expert-curated models (*model zoos*) [28, 33] and domain-specific pre-trained toolkits [20]. These provide ready resources for a user looking to re-purpose components for their own use. However, the process itself can be conceptually challenging because it involves a number of interdependent sub-tasks such as identifying transfer candidates, transferring them successfully, evaluating performance, and identifying next steps. Each can require considerable expertise in the ML model design process. In order to assist domain experts in using components of expert-curated models and leveraging their unique expertise, one must develop workflows which support these tasks and integrate feedback systems which help them to apply their skills. For instance, a mobile application designer seeking to identify sensitive posts before they are submitted might be able to select and tune components of an expert-curated text classification model (potentially trained on terabytes of data) with guidance from their unique understanding of their user base.

In this paper, we investigate the design of interactive environments that support transfer learning. While transfer learning is applicable in a variety of ways across different ML models [53], we focus on Convolutional Neural Networks (CNNs)[18]. ML models

CHI '21, May 8–13, 2021, Yokohama, Japan

ACM ISBN 978-1-4503-8096-6/21/05…$15.00
https://doi.org/10.1145/3411764.3445096

such as CNNs particularly benefit from Transfer Learning since their components often specialize (e.g. identifying object boundaries, colors and shapes in an image [7]), and those specialized areas can be transferred effectively from performant models into new ones [43, 58]. While CNN-based models have complex architecture, their fundamental building blocks are collections of simpler computational units that learn weights. For example, a collection of *filters* (mathematical term for a computational unit) in a model can be exposed as a *layer* that has learned specific features, perhaps better matching users' intuition about transfers.

Through expert interviews, prototyping, and lab studies, we seek to advance our understanding of how individuals conceptualize and apply transfer through interactive tools. In the first part of this paper, we introduce a fully interactive tool which allows users to build new CNN models using a *building blocks* metaphor by transferring components from pre-trained models. We use this as a lens to explore how individuals make sense of the transfer learning process in a lab study and gather both qualitative and quantitative data on how non-experts execute the task using the tool. Finally, we use our lab study data to consider how individuals integrate their own knowledge into the transfer process and identify broader design implications and trade-offs for iML tools for transfer learning. The main contributions of this work are:

(1) A test-bed for gathering and studying user behavior when performing interactive transfer learning tasks. The fully interactive prototype enables building and evaluating CNN models without expert supervision.
(2) The results of a lab study examining information seeking behavior of non-experts in transfer learning. Findings uncover a range of data-driven and perception-driven strategies.
(3) A conceptual model extracted from our empirical data, which points to specific design dimensions for systems where pre-trained models act as *building blocks* for new models.

Our findings suggest that while non-experts are able to conduct transfers across models successfully by employing their domain expertise, their progress is frequently impeded by inaccurate perceptions of the machine's learning process. Through our work, we identify how these perceptions might negatively impact the process in subtle, pernicious ways, and identify strategies for mitigating these risks in the development of future systems.

2 RELATED WORK

Interactive machine learning (iML) tools help users to accomplish tasks at various stages of the ML model building pipeline [46]. iML systems have been investigated for a number of different tasks, including improving input data quality [5, 6, 12], building and analyzing models [9, 47], interpreting model results [30, 44, 49], and evaluating final outcomes [2] (see [29] for a deep survey of related tools). The majority of prior research, however, is geared towards ML practitioners and/or tackling domain-specific challenges. Users with limited ML expertise have received comparably less attention. This research explicitly focuses on the challenge of designing iML tools that empower non-experts to apply ML in their workflows with minimal traditional resources (e.g. terabytes of training data, computational clusters). For the purpose of this paper, we define non-experts in the context of ML model building as *individuals having minimal or no formal knowledge of the machine learning model building process.* These individuals may or may not have programming experience, and can come from a variety of backgrounds including domain experts who might benefit from employing ML in their workflows (e.g. healthcare practitioners, user experience designers), citizen scientists (e.g. distributed environmental monitoring), and DIY enthusiasts (including ML enthusiasts).

A body of research has identified how this community of individuals might benefit from applications of ML [4, 35], and when given the opportunity, can effectively employ domain knowledge in building better ML systems [48, 52]. As a result, a number of projects are investigating ways to support this broader audience, ranging from interactive interfaces [19, 41], accessible and extendable programming toolkits [20], and do-it-yourself (DIY) systems [10, 26]. For instance, Google's ML-Kit [20] is a machine learning suite designed specifically to support mobile application developers and incorporates a bundled library of ML models that can be adapted for unique applications [21]. Designing these ML tools is challenging and requires detailed understanding of the intricacies of the ML model building process and systematically identifying key ML tasks that can be intuitively exposed to the users. Exposing all operations may overwhelm users and steepen the learning curve, while exposing only select features risks limiting the tool's efficacy. In the extreme case, this might confine users' roles to input data providers or performance data collectors. One approach towards striking a balance between too few and too many affordances is to adopt a human-centered design methodology in designing iML tools [25, 56, 57]. In this research, we build upon this body of human-centered design investigations by examining how a building blocks metaphor supporting transfer learning techniques, might allow users to leverage their unique expertise without being overwhelmed by the need to understand intricate model building functions.

Transfer Learning [51] is a widely used technique in the ML community. It is defined as the process of *transferring* knowledge learned by an agent on one task to another, related task. This technique is very useful in building performant ML models with scarce resources. The advent and widespread use of data-driven models like CNNs (which particularly benefit from transfer learning [37, 58]) has further contributed to significant advances in the state-of-the-art in the past several years [45]. A number of recent survey papers have outlined a variety of transfer techniques and principles [40, 53, 60]. Realizing its benefits, the ML community encourages transfer learning by open sourcing their models through model repositories [1, 28, 33]. While these *model zoos* help to advance the boundaries of industry applications and research, we view them as a potential means to bridge the gap between expert and non-expert users via transfer learning. With this perspective, designers may provide richer and more meaningful experience to non-expert users. Prior research [54] has shown that, when done right, iML tools can encourage users to use ML models in unique and useful ways. However, these tools do not account for designing to support transfer learning. In this research, we build upon the emerging line of work and explore the design of iML tools specifically in context to transfer learning. In the future, guided by this research, a practitioner might identify potential models from these pre-existing expert-curated repositories and adapt them to related tasks using their unique domain expertise.

To design iML tools that support interactions with expert-curated models, we adopt a human-centered approach. Several user studies have been conducted to investigate how individuals and teams build ML models [57], select features [55], iterate over data [24], interpret results and explanations [3, 36], and integrate their domain expertise [8, 10]. For instance, one study [8] suggests that individuals tend to use their own perceptions of image data in order to better predict the likelihood of failure of an ML system. Besides focusing on how individuals perceive and adopt the ML building process, some user studies [14, 54, 56] have also investigated the broader challenges faced by specific user groups in working with ML. For instance, [14] identify that UX designers, even after working with ML professionals, have difficulty in grasping the algorithmic part of ML. Through meticulously designed surveys and interviews, these studies draw on the experiences of ML and non-ML professionals.

To build on this body of research, our inquiry is directed towards model building through transfer learning by creating an environment where all of the basic building blocks of the model design process are provided to the user. The main challenges, therefore, are not necessarily in making sense of processes such as feature selection, but rather in understanding what to transfer, how to conduct the transfer, and in what way to evaluate the result. We apply a similar mix of qualitative and quantitative methods, but focus primarily on users' strategies and conceptual model of the task as they are potentially the most likely to benefit or the most likely to fall short in the case of transferring model elements.

3 DESIGN PROBLEM

Before developing our interactive machine learning tool, we worked to identify the specific design challenges posed by such tasks. To formulate our design problem, we investigated the transfer learning process using a mix of expert interviews and literature review. Building on these findings, we then created a preliminary prototype and conducted pilot studies that captured the real-time functional challenges of the iML tool. Using these data, we re-designed our system and then conducted our formal quantitative and qualitative lab study investigating user performance and decision-making.

3.1 Understanding the transfer process

To ground our understanding of the transfer learning process, we conducted semi-structured interviews with 6 ML practitioners (2 graduate students, 2 industry experts and 2 professors). Since a considerable body of literature exists on how ML practitioners build ML models in general, we focused our interviews specifically on transfer learning with ML models, examining the specific steps and stages they considered in the process. Interviews were conducted using open-ended targeted questions like "Describe how often do you employ transfer learning in your model building process," "Please describe any generic approach you take to transfer components across related task," and "Describe any significant challenges you encounter and the specific techniques you use to address them." We specifically focused the conversations on their experiences of building CNN models through transfer learning. The interviews lasted about 45 minutes. The range of expertise of participants helped to expose overarching workflow patterns in transfer learning. In order to get more insight into the specific methods adopted by researchers, we scanned key literature and identified typical transfer learning workflows, generic CNN model building workflows, and how the ML community builds successful transfer techniques. Our methodology here was to identify a central set of influential papers to function as the seeds of our search (cited in our literature review; e.g. [40, 53]). Based on keywords in these papers, we then followed citation chains and conducted targeted searches. Though we did not intend this to be an exhaustive search as in a formal literature review project, we found that these papers provided a good snapshot of relevant issues.

Our findings suggest that there is no standardised transfer technique used by practitioners for CNNs [38, 43, 58], making model building through transfer learning a *highly iterative* process that requires constant assessment and adjustment. Our discussions with practitioners revealed that they often rely on their own intuition borne from years of *experience* and cannot necessarily verbalize why they follow a specific process. For instance, practitioners referenced having a *sense* of the size of filter to use, depth of layers to use, and the hyper-parameters to fine-tune to ensure a well converged model. This reliance on intuition may also be attributed to the black-boxed nature of CNN models [11], which has received considerable attention in research community [23]. The practitioners also pointed out reliable *diagnostic techniques* that helped them to interpret the behavior of CNN models. They further emphasized that users rarely come upon ideal models without a high degree of initial *trial and error* by using various diagnostic approaches, suggesting that providing a tight loop between model design and evaluation is crucial. One expert pointed out that viewing *contradictory evidence* (e.g. exploring both classifications and mis-classifications together) can provide valuable insights, which suggests towards a need for intuitive diagnostic tools. Another practitioner revealed that often the model results felt puzzling at first and deeper investigation of both source and target data samples provided clarifications. For instance, observing differences in model responses for two similar (but not identical) samples helps to identify problems.

We used a qualitative coding methodology to identify the overarching challenges and salient elements of the mental models of our participants, focusing on the specific process commonalities shared between them. These elements are summarized in Figure 1 (b). We found three central patterns shared among participants: *Selection*, *Assembly* and *Diagnostics*. We also coded them by the nature of the task, identifying *Functional* (challenges faced by the user in operationalizing a task) and *Conceptual* (challenges faced by the user in interpreting candidates and results) elements.

3.1.1 Selection (S). The first step in the transfer learning workflow is to identify high-level (data set and models) and low-level (model components) transfer candidates. This requires overcoming the functional challenges of selecting source datasets (*S1*) that might be similar to the target dataset (*S2*), selecting models (*S3*) from the 'model-zoo' as source model candidates, and identifying source model components (*S4*) that can be re-used for the target task. Additionally, the user also needs to address the conceptual challenges of generating potential hypothetical transfer mechanisms (*S5*) and correctly interpreting the model's learned representations (*S6*).

3.1.2 Assembly (A). Once a potential transfer mechanism scenario between a source and target model is identified, the user then needs

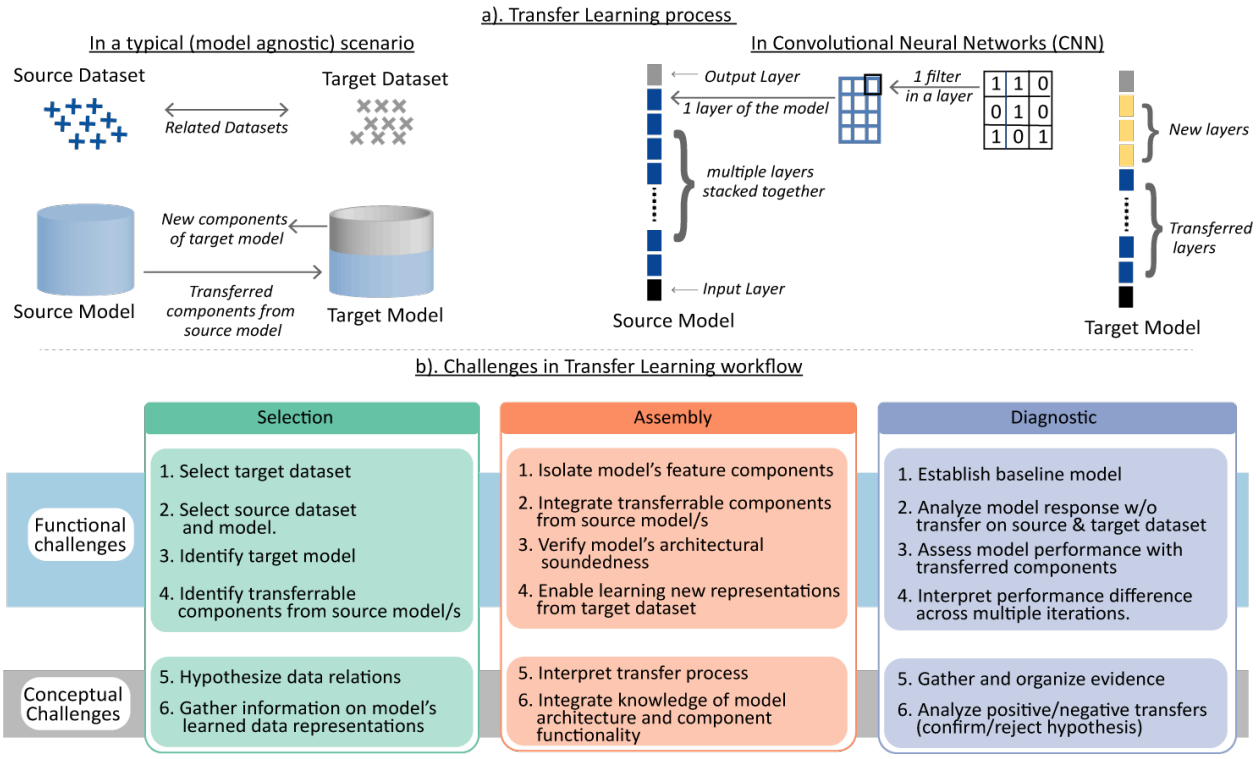

Figure 1: a) The transfer learning process in ML systems (model-agnostic) and in CNN models (model-specific); b) a systematic view of challenges as faced by experts at various stages of transfer learning workflow.

to operationalize the transfer. This includes building a functionally valid model (*A3*) from the building blocks, identifying different pieces of the source model (*A1*), and fitting pieces correctly into the target model (*A2*). For instance, in case of CNNs, the transfer is effected by copying the numerical weights learned by the source model into the parameters of the target model. These functional challenges also include training (*A4*) on target datasets and controlling whether the transferred components should also learn from target task or not. The user also needs to address the challenges pertaining to interpreting the transfer process (*A5*) and develop an understanding about the task execution process of the model (*A6*) in order to asses impacts on the target model.

3.1.3 Diagnostics (D). Diagnosing the success and failure of a transfer mechanism is central to the iterative nature of the transfer learning workflow. In this stage, the user establishes baseline model performance (*D1*) and assesses the model's response (*D2*) both in isolation and in comparison with the baseline (*D4*). Inspecting instance- and class-level performance (*D3*) using appropriate metrics are some of the functional challenges faced by the users during diagnostics. Performance is often evaluated using more than one technique, making organizing evidence from different diagnostic methods important for decision-making (*D5*). Defining acceptable performance for the model in context to the target task and analyzing the performance accordingly (*D6*) are some of the conceptual challenges faced by the user in this stage. For instance, when building an optical character recognition model for an image-to-text converter application, the user may need a higher level of performance for frequently occurring words or characters (e.g. 'e'), than less frequently occurring characters (e.g 'z'), to meaningfully improve the overall model accuracy. These domain specific decisions guide the user towards hypothesizing better transfer mechanisms.

Based on the above findings, we designed a prototype that reflects the intersection between the functional (data, model, visualization) and the conceptual (insights, hypotheses, actions) attributes, allowing different operational paths during iterations. We hypothesize that a system that reflects and nudges users towards expert behavior will help non-experts better adopt the transfer learning workflow.

3.2 Task, CNN Model and Dataset Selection

In order to construct an iML prototype, we targeted a single use case rather than attempting to build a general purpose system. We chose a simple task, English handwriting classification. This classification task was selected primarily because unlike other challenging tasks like scene/object detection, ML experts have created models that are highly accurate (99.99%) in classifying English handwritten characters. As our goal is not to deliver functional improvements to handwriting classifiers, choosing a case that is "solved" guarantees that we will be able to provide users effective functional units to transfer. Further, the task lies at the intersection of vision and text classification – two common applications for ML models among non-experts. Finally, we could reliably recruit domain experts for this task, as participants were very likely to possess fundamental

knowledge of how they identify characters when reading and ought to be able to verbalize their process. This permits an exploration of the strategies non-experts employ when trying to build models using transfer learning 'building blocks'. Additionally, the visual simplicity of the task made sure that users were able to participate in the study without getting lost in intricate details of the dataset. We selected the EMNIST dataset [13] that contains images of handwritten uppercase English characters (A-Z), lowercase English character (a-z) and digits (0-9). The image size is small (28x28 pixels) and hence faster to train with a shallow CNN model. A typical transfer learning process of CNN has been described in Figure 1 (a). The CNN model we used for our study was a simple 4 layer shallow model (grey rectangles in Figure 2 (a)).

4 SYSTEM DESIGN

Following our initial investigation, we integrated the requirements identified in the formative data gathering phase into a prototype tool which supports interactive transfer learning. We use this system to understand the feasibility and efficacy of transfer learning in non-experts, probing how individuals conceptualize the process when the basic building blocks of the model design process are provided. Our goals are two-fold: First, we seek to provide insight into the sense-making process of non-expert users as they work through a transfer task in a lab setting in order to highlight the challenges they face and the strategies they adopt (and consider whether they are unique to this user base). Second, as in a traditional user-centered design inquiry, through the prototype and evaluations we seek to expose specific problems that iML designers need to focus on when building systems for these target users. We did not seek to create an optimal tool that was production-ready or general purpose. Rather, we focused on teasing out the most salient components for a transfer learning system and encoding those into a working prototype.

Our initial prototype focused on implementing the features identified during interviews. We then conducted 3 rounds of pilot studies with multiple participants to validate that the prototype was generally usable. Through this formative investigation, we identified several usability issues such as the absence of guiding messages, too small interface elements, and confusing button organization. One prominent issue experienced by all users was a lack of clarity concerning what to do when in the interface. Users did now know how and when to move between between different stages of the process (an insight that did not emerge during the expert interviews). We redesigned our interface to incorporate a pagination metaphor which reversibly directed users through steps.

Our redesigned test-bed prototype was implemented as a paginated dashboard that nudged users through the transfer task (but ultimately permitted any workflow they desired). The tool is web-based, employing Javascript, d3.js, convnet.js [32], HTML/CSS, and Python in the back-end. The interface provides users with 3 pre-trained models for handwritten uppercase and lowercase English characters and Arabic numerals. Users also have access to individual samples, training, and test sets for each of the 3 datasets. As we did not employ GPU acceleration, training and testing times are higher than would be typical in a final system, but not so high that they are overburdening. We used the general workflow pattern shared among our formative participants (construct, train, test, [transfer, test,] investigate) to organize the interface. While it gently guides the user, they are also free to deviate from steps at any time. The final, post-redesign version is depicted in Figure 2 incorporating the following features:

Creating models: To build a model, users drag and drop using a linked building blocks metaphor (transfer challenge *S3*). In a real-world setting, there are several different parameters to set and tune in order to create a new CNN model. To simplify the task given limited experimental time, we automated the setting of these parameters in the back-end, allowing users to control only select filter sizes, learning rate and sample sizes as part of assembly (*A1*). In the future, these settings might be tuned as an 'advanced' feature or automatically approximated. The models created by the user are also automatically verified for functional/architectural correctness, and the system issues notifications if the user accidentally placed incompatible computational units together (*A3*).

Transferring elements: The same building-blocks metaphor which helps users to assemble a model also allows them to transfer elements from pre-trained models using drag-and-drop from a storage tray, simulating a 'model zoo'. A user can select individual components (*S4*) or an entire model (*D1*) as transfer candidates. To help users keep track of what they transferred, all of the layers are color-coded by source model (*A2*). By looking at the color of layer, the user can tell from which model it has been transferred.

Training models: Users can configure their training to include as many samples as desired, and identify particular class(es) they want to train (e.g. only train on 'Z'; *S2*). The users can opt to prevent certain computational units from learning using a button (*A4*). While in practice a training tool ought to allow deeper levels of configuration (e.g. allowing the user to select loss function, batch size, number of epochs, etc.), we automated the selection process of these features to help users focus on high level tasks. In future versions of our work, we envision a recommender system that suggests these parameters based on user workflow.

Testing models: Users test models via a configuration panel similar to that of training. The users can choose the dataset (*S1*), classes, and the number of samples on which to test. The interface provides a real time view of instance-level classification as it occurs through a bar chart. This is useful to catch places where a model often makes mistakes. When testing is complete, a final accuracy number is displayed at the bottom (*D4*).

In our formative work we identified that experts manipulated input and model components in order to identify the "role" of transferred components (e.g. picking out loops in letters) and to probe the efficacy of a transfer. Therefore, we focused on providing several affordances for exploring the input-output space of the model.

Probing with inputs: Users can select samples from the test dataset to analyze model performance (*D2*). The interface also allows them to generate their own data samples (in this case, through sketching) to observe model performance under specific scenarios– a feature that helps users develop an intuition about their model's response to various stimuli (*S5*). The sketched sample is input to the model and results are presented in terms of confidence among classes (e.g. 60% l, 40% i). Users can also use an occlusion tool to hide portions of a sample from the model (e.g. eliminating serifs; *S6*), generating variations within a sample.

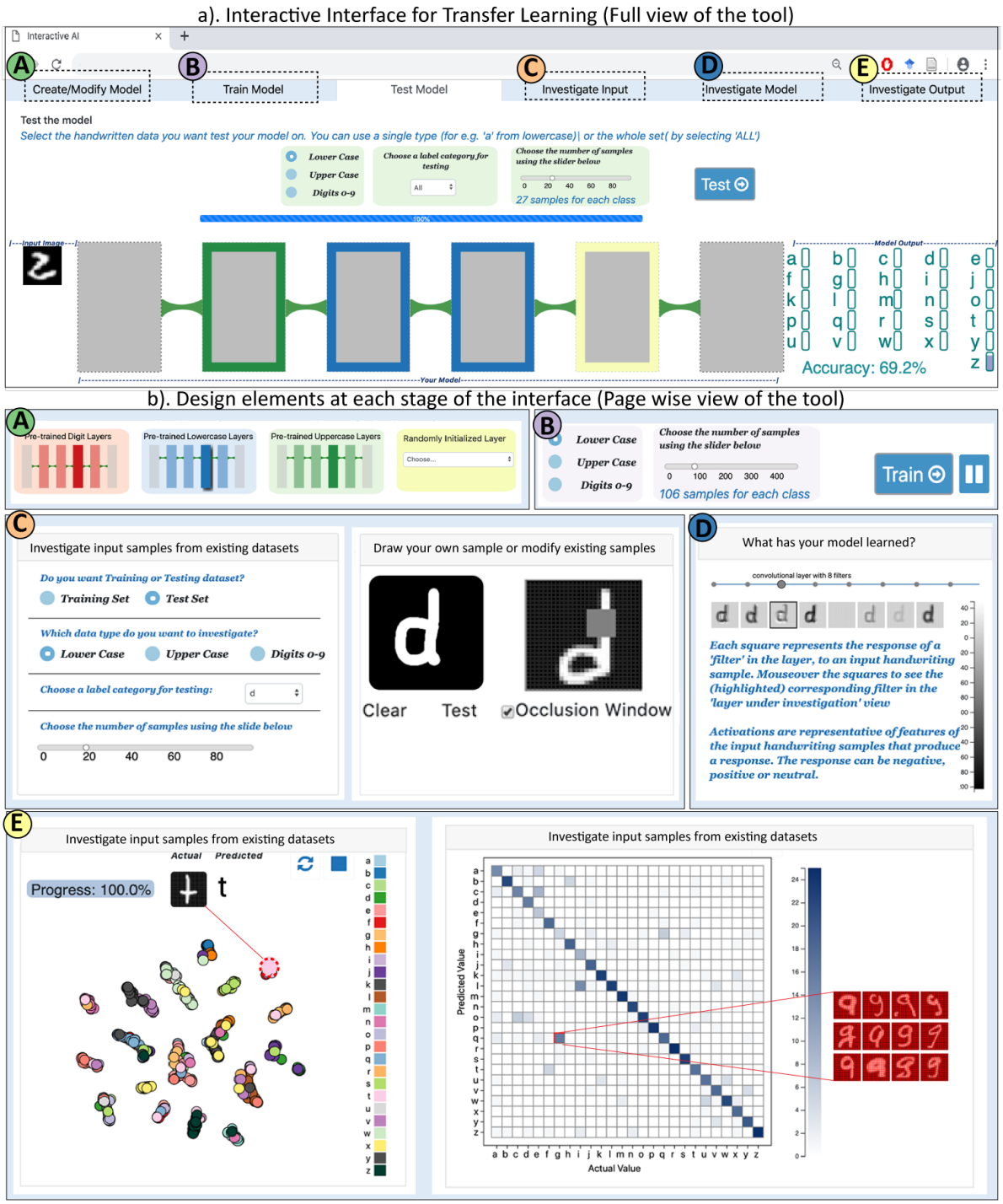

Figure 2: a) The prototype interactive transfer learning system (screenshot shows the model accuracy on overall lowercase and confidence on specific instances of 'z' using a bar chart). b) Features provided at different stages include: i) pre-trained models simulating a 'model zoo'; ii) selecting training datasets using drop down (learning hyperparameter settings are automated); iii) selecting specific input training and test samples to investigate and modify; iv) visualizating models' learned representations; v) exploring model response for the dataset (scatterplot) and individual classes (matrix plot).

Investigating model components: Users can visualize intermediate layer activations and filters' learned representations (*A5*) in order to make better sense of the actual structure of the model. The prototype shows z-scores of each filter (*D3*). Deviation from mean is displayed to highlight rapidly changing filters. This view is linked to the input probes, helping users examine their model in detail. We also provide activation heatmaps [59] (*A6*).

Investigating output: The transfer interface presents model performance through a clustered view and a confusion matrix. The clustered view uses t-SNE [39] to show class-level clusters onscreen in a pan/zoom scatterplot (*D6*). The confusion matrix can highlight small, per-sample errors which (despite their small scale) might be indicative of pervasive issues in the model. Both views are presented side-by-side (*D5*).

5 METHODOLOGY

To further our understanding of how non-experts employ transfer learning to build functioning ML models, we designed a lab study using our prototype. We conducted the study with participants from diverse backgrounds having minimal or no experience with ML systems and demonstrating specific characteristics: domain expertise to contribute (their knowledge of reading and writing) and a desire to learn about ML and use it to complete a task (a side benefit of participant self-selection to engage in the study). These two characteristics match that of who we would expect to use a transfer learning tool in a real life scenario (e.g. a designer integrating a sentiment recognition tool into their app). We focus on several operating questions in the study in order to understand how well participants can perform transfer learning and their behavior. When given the necessary building blocks in a scaffolded and interactive environment, a) can non-experts successfully adopt expert-curated ML models for a given task through transfer learning?, b) how do non-expert users reason about the transfer process?, c) what strategies do users adopt in order to conduct a transfer?, and d) what patterns do non-experts follow over multiple transfers?

Participants for our study were recruited from a university pool. Our participant pool consisted of individuals who had no formal training in ML, coming from a variety of research university fields. While there was homogeneity in terms of age/education, as might be expected in a university pool, we worked to make sure that participants belonged to a variety of domain backgrounds (e.g. biology, communications, design). We specifically recruited this group as a base case of participants who had fewer preconceptions and room for growth. Further, these individuals may be uniquely situated to leverage ML in the future in their own domains. After consenting, participants provided demographic information and reported their experience with programming, machine learning and neural networks via a pre-survey. We evaluated the participant's ML expertise using this survey, and allowed only people who did not demonstrate any prior knowledge of ML topics.

Due to lack of prior background, we specifically included training materials at the start of sessions. Participants received an outline of ML models and how they work both in verbal and print form. Participants were provided time to ask questions and had 10 minutes to get familiar with the tool. Following training, participants spent 60 minutes completing the study task. Participants were encouraged to verbalize their thoughts following a think-aloud protocol. After completing the task, participants reported their experiences in a post-survey, first completing the NASA Task Load Index (TLX)[22] and then answering qualitative free-response questions. Finally, experimenters conducted a verbal debriefing using a semi-structured protocol. We monitored all participant interactions with the prototype through a logger that also stored all of the models created by participants during their session as well as the data/datasets used to train and test them. Studies were conducted one-on-one with an office Windows 10 PC (in an effort to simulate a realistic context).

Participants received the following problem: *Design a model that can classify, as accurately as possible, handwritten samples of lowercase alphabetical characters* (gathered from EMNIST dataset [13]). The participants had access to the prototype's 'model-zoo', that stored expert curated pre-trained models (lowercase English alphabets (a-z), uppercase English alphabets (A-Z), digits (0-9)) as potential sources. They, however, did not have access to the models' training dataset. Optimal performance in this task was not in fact achievable by copying the entire lowercase model in isolation. As training materials suggested, participants were reminded that though they could test an entire pre-trained model to learn about it, they ultimately had to mix and match. They were allowed to make as few or as many models as they liked in order to achieve the highest accuracy possible. We left the task open-ended because we did not want to prime participants with a particular strategy. As our formative work showed that experts often use intuition, we wanted to see how participants made sense of transfer as they experimented and where they applied their domain expertise.

6 EMPIRICAL FINDINGS

In total, 17 individuals participated in the study (of which 10 identified as female and 7 as male). None of our participants reported knowledge of CNNs or transfer learning. All participants reported as being in the 18-25 age bracket. 3 participants claimed to be post-graduate students and the other 14 current university students. During the study, 2 participants did not complete the task as they refused to follow the study protocol and instructions. We excluded their partial data, leaving 15 final participants.

We focused our analysis on examining the overall success of participants in the study in designing models, participants' conceptual understanding of transfer, and how they executed the task. We used the event log data to corroborate some of the claims made by the participants during the think-aloud process. In this section, we first synthesize the qualitative findings from a study conducted with 15 participants and then provide analysis of event-log observations in context to these findings. Subsequently, we introduce a conceptual model grounded in these observations that provides insight into the broader workflow and interaction considerations for this type of iML interface.

6.1 Qualitative Observations

We recorded observations and think-aloud comments during the session as well as during the semi-structured debriefing interviews. Along with self-reported experiences from the post-task survey, we used an open coding methodology to organize these observations and reports. As we coded, several representative themes emerged:

Strategies in identifying transfer candidates: Participants generally employed several different strategies for identifying which elements to transfer at various iterations of the process. All strategies required the participants to make sense of the data and model, generating hypotheses about what kinds of transfers might be fruitful and what the results of a transfer would be. While some transfers were exploratory in nature (i.e. based on the notion "what would happen if I do this"), many transfers were driven by a bottom-up approach, integrating the participant's understanding of the process and their own domain expertise.

- *Identifying targets through task similarities*: Participants identified transfer candidates from models trained on same (lowercase) or closely related (uppercase) datasets. For example, P11 commented "...It feels like digits do not have much to do with lowercase but uppercase looks a lot like lowercase, so maybe I can use that". P11 continued,"... I am not sure how this thing works though, (but), maybe it will combine the images of uppercase and lowercase together in order to generate good response", pointing that relationship between different datasets was perceived as a leverage point. P14 also employed the strategy, thus commenting,"...I want to see if knowing different type of uppercase values can result in better performance". Every participant incorporated this strategy at some point in their session. These task similarity relationships were not only used to increase the model's learned features , but were also used to inspect how robust the model was. For instance, P5 commented, "I wanted to see if it counts any of the uppercase as part of lowercase."

- *Identifying targets by probing the model*: Participants scrutinized features learned by the pre-trained models in the 'model-zoo' by observing their response over different samples using the investigation tools. For instance, P12 made use of occlusion to infer model behavior, reporting, "...with [letter] 'n', when I occlude the lower bottom, I see it [model] thinks its a 'z', which is bad..." The participants probed the model using custom as well as existing samples to identify the precise features correlated with an outcome. For instance, P4 sketched 5 different variants of *a* to understand why their model confused it with another character. P12 spent significant effort in isolating samples that produced the "most accurate" response, while P7, after spending considerable time evaluating model responses on existing samples concluded that transfer may not work at all (P7:"...maybe the pre-trained ones have some sort of bias towards a particular set, so I will just create something that sees everything fresh"). This strategy suggested that participants relied on tools for probing models as a mechanism for testing hypotheses and making inferences about layer/model behaviors they observed.

- *Identifying targets using domain knowledge*: As the tool offered capabilities for testing the assumptions participants had made about task similarities through probing, they often were a means for integrating their domain expertise in the task. Participants specifically used their own domain knowledge of the dataset - handwriting - to inform different transfer techniques. For instance, they focused on variations in how lowercase and uppercase letters are drawn, using those differences to guide a transfer process (P16: "If there is a more weird looking character then it may be uppercase style and may get identified using the green layer...", P11: "I am thinking of the different styles people write. So by introducing the digit style, it might ... give it more information..."). They further sought to employ their understanding of context to recognize potential transfer scenarios. For instance, P12 reported "...if I see [an] adjoining letter to the confusing ones, I can make sense... For instance, my *q* followed by *u* is likely *q* and not *g*..." and hypothesized that it might improve models further. Participants also requested to eliminate perceived erroneous samples that might be "messing up" the system (P1: "... if you draw *b* and *a* without a long tail, it looks like a *0*"). This suggests that our participants readily made use of their unique domain expertise, so long as they had input tools which allowed them to encode their knowledge into probes (e.g. drawing letters).

- *Identifying targets based on understanding of model architecture*: Participants frequently used their understanding of the model architecture (albeit accurate or flawed) in order to complete the task. They asked questions about what each component meant and how they could alter it. These perceptions of the model architecture were key drivers in identifying potential transfer candidates. For instance, P7 perceived the stacked layers as some sort of logical flowchart where each layer performed a designated task, reporting, "...Its seems to me that by putting yellow [random initialized layer] towards the end... it might only learn *some* new information...", P9 perceived the filters as *computational units* and justified their hypothetical transfer by reporting, "more computation power means more units doing the same job, .. I should definitely assign more computation towards the end and less towards the beginning..." and P15 designed all models based on the perception that layer positions have an important role in how model behaves (P15: " I think maybe position of layers is affecting the accuracy of the model, ... like [it matters] where I put the blue [lowercase] and green[uppercase] layers",P16: "I can see there were more filter in other layers, maybe they learnt from uppercase..."). These observations suggest that that our interactive interface provided affordances which evoked these crucial transfer learning concepts that were identified during the preliminary data gathering phase.

However, while these affordances helped users hypothesize about transfers dependent on model architecture, they did not provide accurate descriptions about the model's learning process. We note that participants' observations were highly sensitive to the ways in which the model was presented to them due to their low level of expertise. The interface and their understanding of the tool shaped how they went about constructing and testing models. In some cases this was advantageous, but in others it brought about risky assumptions that were not grounded by observed evidence.

Role of perceptions of the learning process: During our data gathering phase, one key requirement was the need to iterate over multiple models through different transfer procedures. In practice, this requires an analyst to train a model several times. We observed that while the participants were able to accomplish that, their specific perceptions of the statistical learning process played a key role in their decision-making. Several participants interpreted model training as an incremental activity using the metaphor of human learning. For instance, when re-training the model with a new dataset, they presumed that it would implicitly retain the learning from previous iterations. The interface explicitly provided a toggle button to control the learning rate of a layer or a filter, giving users the authority to decide if they wanted to retain the learned parameters in the model or not. However, the participants rarely used this button because of their use of a human learning

metaphor, and it often led the participants to perform iterative training sessions with insignificant results. Participants selected specific samples to "show" the model, with the assumption that the training process will not change the model's prior learning (P12: " it doesn't really need to re-learn the same information...", P14: "I want to see if it remembered anything from last time", P2:"...if I train each letter individually, then it should perform better"), and incrementally trained their model to control what it learned (P4: " I hope there was a way to train more specific samples, I don't want (to) just train the whole thing"). A prior study with UX designers [14] suggested that the difference between statistical notions of intelligence and human intelligence may lead to conflicted interpretations of machine results. In this research, not only did we observe the impact of this interplay during the users' decision making process, but also observed that misinterpretations of the statistical learning process can pose significant barriers for non-expert users. While this is certainly an unexpected usability concern for our tool, more generally this is an issue that merits consideration for non-expert users engaging in test/train iterations.

Adopting techniques for tracking progress: Participants often had to recall models that they had previously designed. Since they were instructed to achieve as high an accuracy as possible, they focused heavily on accuracy numbers in comparing past models to present. A common strategy amongst all participants was to keep track of the model that did best on a given dataset and try to work from there (P9 "I think the best combination was the one that I made before") potentially even recreating it if they wanted to backtrack (P1: "Maybe introducing digits was a bad idea. Okay, so I will revert back to the old model and train on lower case higher number of samples"). In general only 1-2 past transfers were tracked by participants, though for more distant transfers participants attempted to retrace their 'cognitive steps'. They backtracked whole models and also often tried to backtrack pieces of prior models and combine them in new ways, leading to a sort of within-model transfer. While the experts did not talk in detail about this activity, the user studies demonstrated that tracking progress and being able to differentiate between models/transfers is crucial for non-expert users who might be experiencing higher cognitive load.

Finding patterns across model outcomes: Almost all participants used the *investigate output* tab to identify a class where the model performed especially poorly (P7: "*b* were never mistaken for *m*, if you filled in that line (pointing to confusion matrix)"). A general technique for participants was to isolate data elements that had poor performance and only focus on them. They also used their domain knowledge to identify data samples that could make up the difference for variations. For instance, P11 hypothesized that *g* is much more common letter than *q* so it [model] may be just ignoring the tilt in the *q*... and some of the *q* look like *9*, .. so maybe I can borrow..." The participants also employed poor strategies such as training only for one class, expecting that the model would retain all the previously learned features perfectly (P12: "I am trying to see how sensitive it is to u's , v's and w's.... ohh. so now it is pretty good at only u's and v's..."). While the strategies they employed were generally intuitive, the execution misaligned with canonical techniques owing to how they perceived the process. Furthermore, 1-2% changes in accuracy were seen by the participant as sufficient justification for a successful transfer technique or failure. The investigation tools were mostly used when the transfer was negative (i.e. the accuracy dropped). This suggests a need to mitigate users' over-reliance on accuracy numbers and incorporate affordances in the tool which help in interpreting them explicitly.

Attributing high importance to size and variation in data: All participants assigned high importance to the number of samples they used. Their first hypothesis often was that training and testing models on higher number of samples for each class would always yield better accuracy (P4: "bigger sample means more variation"). Every participant tried to adjust sample size to achieve higher accuracy and wanted to enhance the dataset by adding many of their own samples in order to inform the model of more variations. The notion of "teaching" the model aligns with the previous finding on concept overlap risks/rewards.

6.2 Observations from usage patterns

We analysed participant system usage patterns using data from 1) an event logger that captured all participant interactions including the models they built, what layers they used, and on what datasets they trained/tested the model; and 2) self-reported survey results. While the survey results provided insights into the demographics and the cognitive load participants experienced while using the tool, the event logger provided insights into their conceptual understanding. We analyzed how individuals' transfer strategies and use of the tool evolved over time through aggregated workflows (e.g. moving from Creating to Testing to Investigating Input to Training and then back to Testing), and time distribution as shown in Figure 3. We visualize these data in a state diagram where each circle denotes one of the 6 steps of the process. Links between states signify that the user moved between them, weighted by how frequently the user moved. We only picked transitions where the user actually performed the task and eliminated random clicks. These state diagrams are shown in Figure 3 (b). We also visualize the models they created using stacked bars, color-coded for the different layers participants chose to use. Although users built many models during the session, we selected only the models that users opted to test/inspect and recorded their accuracy.

In examining log data, we noticed several important patterns participants employed. We identified these patterns by tracking the models they designed in the Create, and Train and Test stages. Our event logger captured every single model involved in all 6 stages. On an average, participants designed a wide number of models ($Max = 14, Min = 4.3, SD = 3.1$). However, some participants had the same models in the Create and Train stages, suggesting a *model-centric* approach, while other participants ended up with significantly fewer models in Create stage than in Train stages thus suggesting a *data-centric* pattern. These are described below.

Model-centric patterns: Some users navigated more frequently between Creating a Model, and Training/Testing (Construct stages). Their information-seeking strategy was to test different variants of models on the same data, occasionally varying the size of training and test dataset. We identified these participants (40%) by their model-building behavior and then studied the distribution of their session time across different stages. We observed that these users spent slightly more time on Investigate stages ($M = 53.01\%$) (Io, Im,

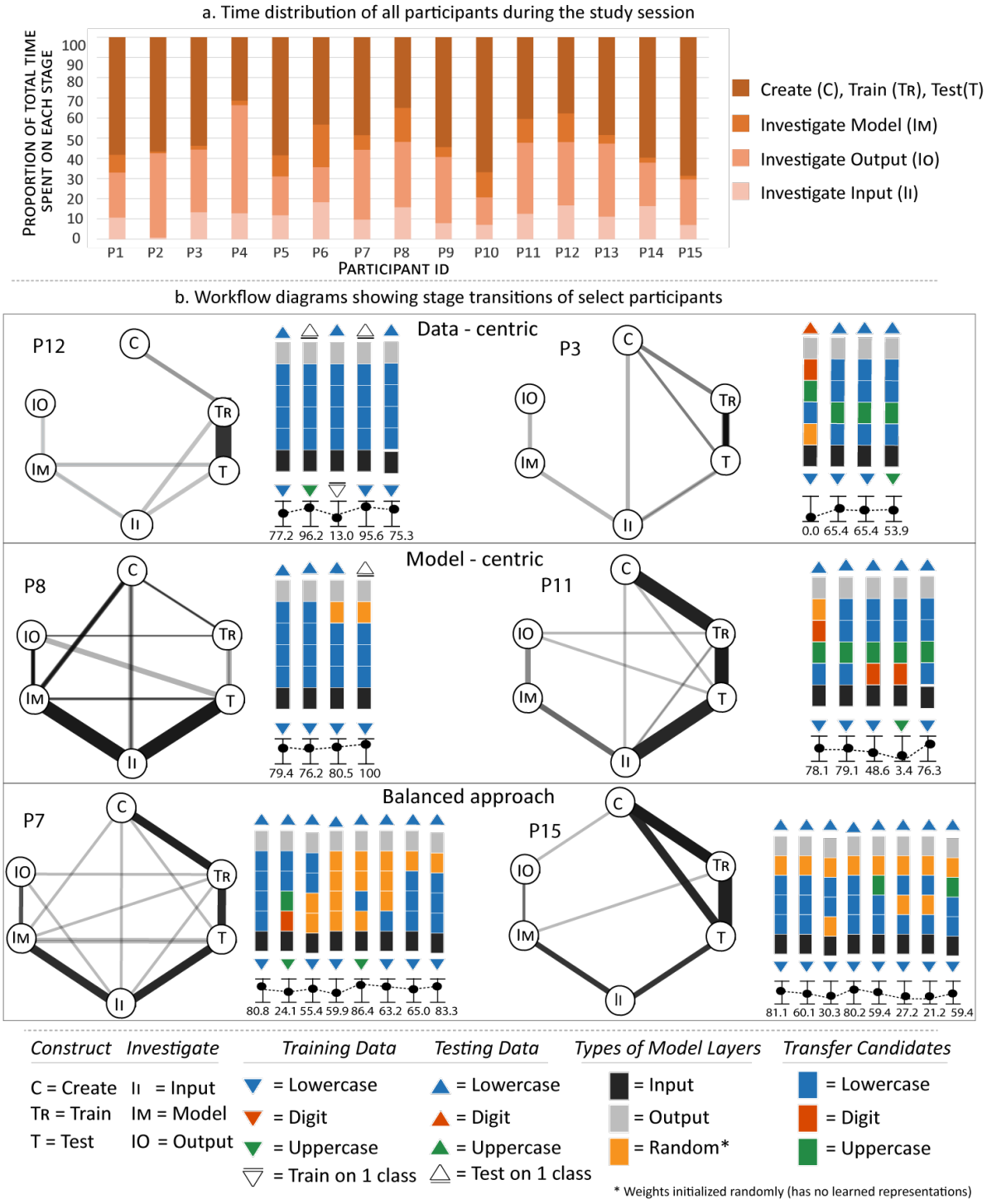

Figure 3: a) Time distribution of each participant and b) examples of workflows adopted by participants for the three patterns observed. The state-flow diagram shows the frequency of participants' movement between the 6 stages. A darker (and thicker) line between 2 stages means the participant moved more frequently between them. Also shown alongside are select models built by the participants (vertical stack of colored rectangles). Each layer in the model is color-coded by its origin. For instance, a blue rectangle in a stack means the participant selected this layer from pre-trained lowercase model. A triangle represents the dataset used to train/test that particular model.

Ii in Figure 3) as compared to the users who adopted data-centric approach ($M = 44.76\%$). However, this difference was not very significant statistically ($t(9) = 1.36, p = 0.20$). The proportion of time spent by both of these user groups was also not significantly different for Im and Ii stages and only marginally different for Io ($t(9) = 1.46, p = 0.17$). When observed closely, this suggests that even though one would expect users practicing a model-centric approach to spend more time creating, training/testing the models, these users might be guiding their hypothesis using information gathered from investigating both data and model.

Data-centric patterns: On the other hand, several participants quickly arrived at a model and navigated frequently between Test and Train stages. Their information-seeking behavior was to try different datasets on the same model, occasionally and slightly modifying the model. These participants (33.3%) often reported relying on their understanding about the data to inform the transfer process. We observed that these users spent a slightly higher proportion of time in Construct stages (C, Tr,T in Figure 3) ($M = 55.24\%$) than the users who adopted a model-centric approach ($M = 46.98\%$), but with marginal significance in the difference between the two groups ($t(8) = -1.34, p = 0.21$), suggesting that investigation tools were proportionately used to inform these strategies. It is also important to note that the time spent in the Tr and T stages was proportional to the size of the dataset selected. So users who frequently trained and tested on large sample sizes recorded a high proportion of time spent in these stages.

Balanced approach patterns: We observed that several users employed a mixed approach where they switched between data-centric and model-centric approaches. The users designed several models by training and testing on same datasets, and then switched to refining select models using different datasets. This strategy is distinct in state flow diagrams (Figure 3(b)). The users who employed this approach generated hypotheses that were based both on model outcome and their interpretations of data relations.

In addition to the observed patterns, we also found that participants spent significantly more time on investigating output than investigating input and models ($F(2, 28) = 32.27, p <= 0.001^{**}$). We also fit a linear model for the number of models designed, trained or tested by the users over time to demographic factors in an effort to isolate any influence of prior technical experience on user performance. Using self-reported gender identity and technical experience as predictors for accuracy (age was not included as it was largely uniform over participants), model fit was generally poor ($F(11, 3) = 0.70, p = 0.57$). We also analyzed the TLX scores reported by the users in the post-survey to measure the cognitive load by experienced by the users. As might be expected, we found that users who designed more models self-reported lower cognitive load ($\beta = -1.03, t = -2.62, p = 0.02^{*}$), indicating that users who felt less overloaded by the tool were more able to efficiently use it to build models.

7 CONCEPTUAL MODEL

In our study we observed that, given the right support, users were able to employ their domain knowledge in successfully transferring elements from expert-curated models into their own models. Considering participant sessions holistically and concentrating on the qualitative themes and event patterns exposed in our data, we constructed a higher level representation of the different information pathways these non-experts pursued when performing transfer learning over multiple iterations. Much as in traditional work on sensemaking [42], we construct this model in order to identify potential leverage points in interactive transfer learning tools for non-expert users. Further, we can use this model as a lens through which to reflect on key points where our participants differed from experts on a procedural or informational level. In the following paragraphs we walk through this model as shown in Figure 4.

Gathering Information: ML non-experts do not come into the task with background knowledge or intuitions for identifying transfer candidates as experts do. Instead, they begin the process by gathering information through experimentation and probing. This is akin to "What-if Analysis" (WIA). In WIA, users capture and study the response of a system under different conditions. In our data, two kinds of WIA strategies emerge: *data-oriented WIA* and *model-oriented WIA*. In data-oriented WIA, a user keeps the model constant and studies its response to various data instances. User strategies pertaining to *probing the model* are instances of data-oriented WIA. In model-oriented WIA, the user keeps the data constant and studies the fit of various model combinations on it. This requires them to employ strategies that incorporate *knowledge of model architecture and functionality* to correctly interpret the results and formulate hypotheses. While both techniques yield richer information, model-oriented WIA has higher costs with respect to time, effort and cognitive complexity. For non-experts these costs may be too high, but experts are more liable to use model-oriented WIA as their cost structure is reduced by experience and training. Key here for interactive systems is to offer tools which reduce the costs of exploring alternative model formulations while providing precise explanations of model architecture.

Hypothesis Generation: After gathering information, participants hypothesize and identify transfer candidates. Not all machine knowledge is transferable, and identifying potential avenues can be challenging. In order to locate a transfer, individuals must synthesize knowledge both about the model as well as about the data. *Data Knowledge* refers to the user's insights and intuitions about the dataset. User hypotheses are often based on data knowledge acquired outside of the ML environment. On the other hand, *Model Knowledge* refers to the user's information about the model pertaining to its architecture, learned representations and working components, which is often acquired within the ML environment. Both of these inform hypothetical transfer techniques. However, as in the previous stage, hypothesizing imposes differing costs and challenges based on expertise. For an expert, data knowledge comes from knowledge of statistical relationships (based on metadata like size, variance, density and complexity) and model knowledge refers to the fundamentals of model design and learned representations. On the other hand, data knowledge for non-experts primarily comes from their own domain knowledge (content, implications, insights, behavior) and model knowledge from specific model compositions that seemed to work in previous iterations. Even though the cognitive cost of acquiring model knowledge is significant high for non-experts, the lack of this knowledge may lead to inadequate use of data knowledge. A potential avenue for enhancement in systems

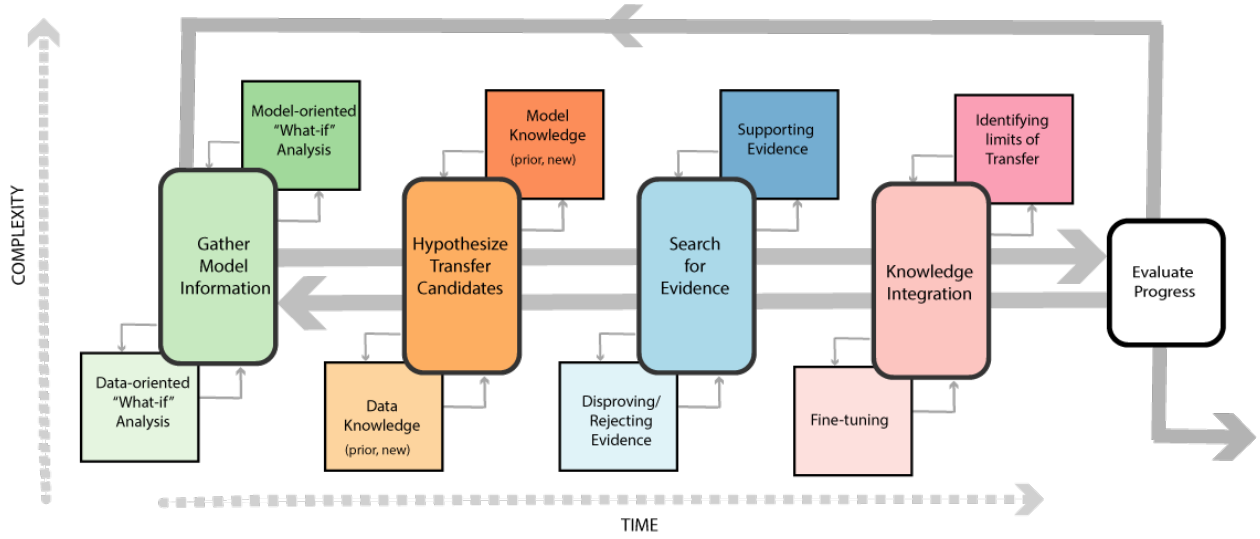

Figure 4: The resulting conceptual model. Lower items have less complexity/costs versus upper items. Iteration happens within steps, between steps, and from the start to the end of the model. Users exit at the right.

is augmenting the gathering of model knowledge and providing candidate recommendations to reduce perceived costs.

Searching for Evidence: Having now transferred their candidate into the model, the user must again gather evidence in order to determine if the transfer was successful (testing their hypothesis). A transfer may be rejected based on negative evidence (e.g. a decrease in model accuracy) or may be supported by confirming evidence (e.g. higher accuracy on elements hypothesized). At this stage, the user ought to explore both sides in order to fully evaluate their success. However, the presence of confirmation bias and reductive-but-easy-to-interpret measures such as test accuracy can cause non-experts to fixate on low-cost, low-information evidence. Strategies relying on probing the model with samples, examining model structure, and observing per-class output are more challenging, but might provide fine-grained insights about performance. For an expert, the costs of these more advanced analytical techniques are reduced. This points towards opportunities not only in making systems that are generally more resistant to analysis bias, but also in drawing users' attentions to other measures of success beyond numeric accuracy through nudges.

Integrating: Finally, the participant must decide what action to take as a result of the evidence they have observed. If the transfer failed and they noticed it, then they ought to return back to the start of the process and make another attempt. If it succeeded, then they must decide whether to loop for more improvements or to satisfice. Key in this stage is gathering evidence to identify when the model is sufficiently performant and *integrating* the evidence found into an action plan. On a shallow level, one might iterate and try something new. Though it imposes higher costs, a strategy of analyzing evidence to identify the key weakness or specific benefit of a transfer might deliver better results by hastening future iterative developments. For experts, practice might make this strategy less costly through intuition. Non-experts, due to higher costs, are more likely to take the shortcut strategy or prematurely end with a sub-par result. Tools that push users to better evaluate model performance and identify next steps may help to reduce costs.

8 DISCUSSION

In this paper we first designed a prototype tool for building and evaluating machine learning models using transfer learning and used it as way to probe how non-experts might use such tools through a lab study. In examining our lab study data, we identified key breakpoints and patterns of use which both aligned and misaligned with patterns reported by expert users. In many cases, lack of expertise shaped the way in which the tool was used by non-experts, and is a key place for consideration in the design of future tools. In identifying general working strategies for our participants, we found how different users fixate on data-, model-, and performance-level factors as they build. For each of these, there are potential leverage points in providing helpful affordances or nudges in the right direction. Thinking more broadly, in this section we reflect on the overarching design issues and trade-offs for iML tools which support transfer learning with an aim towards issues which generalize beyond transfer learning interfaces into general ML tools for non-experts.

Mitigating misaligned learning perceptions using design: In the early stages of our interface design, our focus was to address the functional and conceptual challenges that non-experts might encounter. The prototype not only provided the building blocks to help in selecting, assembling and diagnosing a model, but also provided a scaffolded environment to guide users in how to use these building blocks. Our study, however, indicated that the perceptions of learning played an important role in how these building blocks were employed by participants. Users' perceptions about how the machine 'learns' were often derived from their inherent understanding of the human learning process. As a result, underlying differences between the two often lead to misinterpretations (e.g. how model training functions statistically). The users also relied

on these perceptions to fill gaps in their knowledge as early in the process as the information gathering phase. When unaccounted for, these misinterpretations bring the risk of users inadvertently encoding values, assumptions, and beliefs that may produce performant models with concerning deficiencies (e.g. bias towards a certain data sample). One possible way that designers might mitigate the impact of these misinterpretations is by incorporating intuitive visualizations of the *declarative knowledge* of the model. For instance, if every training iteration in the model erases its prior learned weights, a visual metaphor should provide the users with explicit feedback demonstrating this behavior. Incorporating the elements of declarative knowledge with the *procedural knowledge* of model design (what steps to take next) into the interface can help non-expert users integrate their domain expertise into ML design process. More generally, there is a need to develop visualizations and descriptions which efficiently convey to users the actual behavior at play, and to avoid assumptions about users' mental model of the ML process.

Designing comprehensive model assessment and tracking: Our study task required non-expert users to design a model that can correctly classify handwriting samples in lowercase English using expert-curated pre-trained models. Users demonstrated competence and applied a number of strategies to assess the model's performance: generating *adversarial examples* to fool the system and test its robustness, dissecting model performance on *ambiguous samples*, and comparing outcome distances amongst different groups of samples. Accuracy as a percentage, however, remained the most-used measure of model performance. Since it is memorable and easy to track across different iterations, it is a convenient target for fixation, to the point that other performance measures such as robustness and generalizability are ignored. Further, this measurement is highly sensitive to misinterpretation. For instance, accuracy percentages compared across two models tested on slightly different sizes of data samples lose statistical meaning. Designers and researchers might alleviate this issue by explicitly encoding baseline measures into the tool. By designing and integrating model assessment features that provide a comprehensive view of a single model (such as (mis-)classification matrices or distribution across feature spaces), cognitive load as well as accuracy fixation might be reduced. In the most extreme case, it may be most efficient to avoiding revealing accuracy to novices at all, and only to make use of indirect measures which are more robust against over-reliance and misinterpretation.

Managing information across multiple iterations: While in some cases users in the study were cognizant of the risks of comparing accuracy over disparate model iterations, they experienced serious difficulty in managing information across transfers, which often led to misguided hypotheses concerning potential transfer candidates. In general, we observed highly iterative workflows across our participants. However, they expressed difficulty tracking their progress over time as complexity rose. While participants often wanted to refer back to previous models, the affordances of our system made it such that they had to recreate past steps. Though this might be alleviated with a traditional history and undo/redo stack, it also points to a deeper question. Participants were readily able to store and recall past instances of model training through the procedures they used and often used such process information in order to contextualize their next iteration. In the future as more complete tools are built, there remains an open question as to how best to surface historical and process data to users so that it aligns with their understanding and goes beyond individual interface steps.

Assessing and mitigating risks of scaffolding tasks: Our interface design focused on scaffolding a conceptually difficult task (transfer learning) for non-expert users. Decomposition of the process into functionally similar steps, schematic visual metaphors shared between stages to promote a consistent mental model, and intuitive feedback structures all worked to help users perform complex transfer learning operations from one model to another. More importantly, these elements allowed users to focus their mental effort on integrating their domain knowledge into the models. However, scaffolding comes with the risk of forcing individuals to follow a predetermined pattern inherently conveyed by the interface. We observed in our study that towards the start of the sessions, users were hesitant to break from this pattern. While this was a useful step in quickly familiarizing users with a *valid workflow*, extensive scaffolding poses risks of railroading users. Prior work in the design community has emphasized how early stages of a prototyping process can cause individuals to fixate on singular, potentially disadvantageous paths, risking poisoning the entire workflow [15]. This risk might be mitigated by providing multi-stage, hierarchical or recommender-driven scaffolding that allows users to arrive at workflows that best suit their needs. Recommender systems that specifically focus on suggesting useful next steps based on the users prior steps in the transfer learning workflow pipeline may assist non-expert users in their progress. It is important to note that one can be scaffolded through both a successful and a *deceptively successful* workflow, and it is contingent on system designers to consider potential adversarial and unexpected situations which non-expert users might ignore or fail to notice. Therefore, it would be worthwhile to explore the nuances of an expert's workflow in transfer learning tasks and identify strategies that evoke those intuitions among all kinds of users.

9 LIMITATIONS

There are a few limitations to consider with regards to our prototype system, study methodology, and analysis. Since, our interface was designed specifically for transfer learning with CNN models, it did not account for the unique nuances that other models might bring. Scaffolding the model design process as well as specific inspection affordances may differ substantially in the case of other models, though we believe general non-expert patterns of use will remain consistent. As our interface was not intended to be production tool, many hyper-parameters were hard-coded with best estimates for our limited task-set and the back-end could only handle shallow, 4-layer models. While this sufficed for character recognition, it would not be suitable for more in-depth modeling and may be another area where non-experts will need specific support. Studying more complex cases might offer richer data, as it would enable even more domain-specific evaluations of transfer by novices (e.g. art historians performing artist recognition) at the risk of narrow study populations and a greater training burden. Additionally, while

images and natural language datasets, as used in this study, may be more intuitive for non-expert users to work with, performing transfer learning with advanced tabular (or purely numerical) data may pose significant challenges. One open question remaining from our limited case is also in how biases might propagate during transfer of more complex models and remain undetected due to user implicit bias (or oversight). While we believe our overarching workflow findings represent a general pattern, task complexity is a threat to generalizability that ought to be studied further.

Our study methodology was again limited in terms of sample, tasks, and duration. Much remains to be learned from more diverse populations of users beyond a university environment, and longer duration studies might expose additional insights from novices or asymptotic gains in task performance. This is an area where a formal contextual design process or activity theoretic analysis could provide deeper insights. While we endeavored to find cross-cutting themes in our qualitative data and align them with log and quantitative results, they may still be subject to biases from peculiarities in our interface, study environment, or analytic approach. In the future we hope to investigate transfer in higher complexity/stakes situations through connections to citizen science. Such efforts could provide access not only to expert use cases, but also eager novices who might want to experiment with ML. To make finding and employing transfer candidates easier, we propose utilizing the extended repositories of models and connect them with interactive tools that help users to identify transfer candidates with a large user base. Finally, we hope to explore how other interfaces (e.g. Tensorflow) deviate from the process model we observed.

10 CONCLUSION

In this paper we designed and evaluated an interactive environment for transfer learning. We gathered qualitative and quantitative data in order to synthesize a conceptual model that identified key themes and leverage points in the users' information-seeking behavior in the environment. Our findings suggest that while non-experts are able to conduct transfers across models by relying on their domain expertise, their progress is frequently impeded by inaccurate perceptions of the machine's learning process. As machine learning models continue to grow in complexity and reach more individuals across the globe, there is an opportunity for new systems to unravel these complexities, allowing diverse individuals to employ these powerful tools in their everyday lives.

ACKNOWLEDGMENTS

We are thankful to Prof. Khairi Reda from Indiana University-Purdue University for providing valuable feedback and insights during the preliminary prototype design phase of this research. We are also thankful to Cornell University for their continued support and funding for this research. Finally, we thank the anonymous reviewers for their feedback that helped to improve this research.

REFERENCES

[1] Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, Manjunath Kudlur, Josh Levenberg, Rajat Monga, Sherry Moore, Derek G. Murray, Benoit Steiner, Paul Tucker, Vijay Vasudevan, Pete Warden, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. 2016. TensorFlow: A System for Large-Scale Machine Learning. In *Proceedings of the 12th USENIX Conference on Operating Systems Design and Implementation* (Savannah, GA, USA) *(OSDI'16)*. USENIX Association, USA, 265–283.

[2] Y. Ahn and Y. R. Lin. 2020. FairSight: Visual Analytics for Fairness in Decision Making. *IEEE Transactions on Visualization and Computer Graphics* 26, 1 (2020), 1086–1095. https://doi.org/10.1109/TVCG.2019.2934262

[3] Ahmed Alqaraawi, Martin Schuessler, Philipp Weiß, Enrico Costanza, and Nadia Berthouze. 2020. Evaluating Saliency Map Explanations for Convolutional Neural Networks: A User Study. In *Proceedings of the 25th International Conference on Intelligent User Interfaces* (Cagliari, Italy) *(IUI '20)*. Association for Computing Machinery, New York, NY, USA, 275–285. https://doi.org/10.1145/3377325.3377519

[4] Saleema Amershi, Maya Cakmak, William 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

[5] Mykhaylo Andriluka, Jasper R. R. Uijlings, and Vittorio Ferrari. 2018. Fluid Annotation: A Human-Machine Collaboration Interface for Full Image Annotation. In *Proceedings of the 26th ACM International Conference on Multimedia* (Seoul, Republic of Korea) *(MM '18)*. Association for Computing Machinery, New York, NY, USA, 1957–1966. https://doi.org/10.1145/3240508.3241916

[6] Jürgen Bernard, Marco Hutter, Matthias Zeppelzauer, Dieter Fellner, and Michael Sedlmair. 2017. Comparing visual-interactive labeling with active learning: An experimental study. *IEEE transactions on visualization and computer graphics* 24, 1 (2017), 298–308.

[7] Alsallakh Bilal, Amin Jourabloo, Mao Ye, Xiaoming Liu, and Liu Ren. 2018. Do convolutional neural networks learn class hierarchy? *IEEE transactions on visualization and computer graphics* 24, 1 (2018), 152–162.

[8] Nathan Bos, Kimberly Glasgow, John Gersh, Isaiah Harbison, and Celeste Lyn Paul. 2019. Mental models of AI-based systems: User predictions and explanations of image classification results. *Proceedings of the Human Factors and Ergonomics Society Annual Meeting* 63, 1 (2019), 184–188. https://doi.org/10.1177/1071181319631392 arXiv:https://doi.org/10.1177/1071181319631392

[9] M. Brooks, S. Amershi, B. Lee, S. M. Drucker, A. Kapoor, and P. Simard. 2015. FeatureInsight: Visual support for error-driven feature ideation in text classification. In *2015 IEEE Conference on Visual Analytics Science and Technology (VAST)*. IEEE Computer Society, Los Alamitos, CA, USA, 105–112. https://doi.org/10.1109/VAST.2015.7347637

[10] 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* (Honolulu, HI, USA) *(CHI EA '20)*. Association for Computing Machinery, New York, NY, USA, 1–8. https://doi.org/10.1145/3334480.3382839

[11] Davide Castelvecchi. 2016. Can we open the black box of AI? *Nature News* 538, 7623 (2016), 20.

[12] Minsuk Choi, Cheonbok Park, Soyoung Yang, Yonggyu Kim, Jaegul Choo, and Sungsoo Ray Hong. 2019. AILA: Attentive Interactive Labeling Assistant for Document Classification through Attention-Based Deep Neural Networks. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems* (Glasgow, Scotland Uk) *(CHI '19)*. Association for Computing Machinery, New York, NY, USA, 1–12. https://doi.org/10.1145/3290605.3300460

[13] Gregory Cohen, Saeed Afshar, Jonathan Tapson, and André van Schaik. 2017. EMNIST: an extension of MNIST to handwritten letters. http://arxiv.org/abs/1702.05373

[14] Graham Dove, Kim Halskov, Jodi Forlizzi, and John Zimmerman. 2017. UX Design Innovation: Challenges for Working with Machine Learning as a Design Material. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems* (Denver, Colorado, USA) *(CHI '17)*. Association for Computing Machinery, New York, NY, USA, 278–288. https://doi.org/10.1145/3025453.3025739

[15] Steven P. Dow, Alana Glassco, Jonathan Kass, Melissa Schwarz, Daniel L. Schwartz, and Scott R. Klemmer. 2011. Parallel Prototyping Leads to Better Design Results, More Divergence, and Increased Self-Efficacy. *ACM Trans. Comput.-Hum. Interact.* 17, 4, Article 18 (Dec. 2011), 24 pages. https://doi.org/10.1145/1879831.1879836

[16] Jerry Alan Fails and Dan R. Olsen. 2003. Interactive Machine Learning. In *Proceedings of the 8th International Conference on Intelligent User Interfaces* (Miami, Florida, USA) *(IUI '03)*. Association for Computing Machinery, New York, NY, USA, 39–45. https://doi.org/10.1145/604045.604056

[17] Rebecca Fiebrink, Perry R. Cook, and Dan Trueman. 2011. Human Model Evaluation in Interactive Supervised Learning. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Vancouver, BC, Canada) *(CHI '11)*. Association for Computing Machinery, New York, NY, USA, 147–156. https://doi.org/10.1145/1978942.1978965

[18] Kunihiko Fukushima. 1980. Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biological cybernetics* 36, 4 (1980), 193–202.

[19] Google. 2019. Teachable Machines. https://teachablemachine.withgoogle.com/

[20] Google. 2020. Google ML KIT. https://developers.google.com/ml-kit

[21] Google. 2020. ML-Kit App Lose It. https://developers.google.com/ml-kit/case-studies/lose-it

[22] Sandra G. Hart and Lowell E. Staveland. 1988. Development of NASA-TLX (Task Load Index): Results of Empirical and Theoretical Research. *Advances in Psychology* 52 (1988), 139–183. https://doi.org/10.1016/S0166-4115(08)62386-9

[23] F. Hohman, M. Kahng, R. Pienta, and D. H. Chau. 2019. Visual Analytics in Deep Learning: An Interrogative Survey for the Next Frontiers. *IEEE Transactions on Visualization and Computer Graphics* 25, 8 (2019), 2674–2693.

[24] Fred Hohman, Kanit Wongsuphasawat, Mary Beth Kery, and Kayur Patel. 2020. Understanding and Visualizing Data Iteration in Machine Learning. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems* (Honolulu, HI, USA) *(CHI '20)*. Association for Computing Machinery, New York, NY, USA, 1–13. https://doi.org/10.1145/3313831.3376177

[25] Sungsoo Ray Hong, Jessica Hullman, and Enrico Bertini. 2020. Human Factors in Model Interpretability: Industry Practices, Challenges, and Needs. *Proc. ACM Hum.-Comput. Interact.* 4, CSCW, Article 068 (May 2020), 26 pages. https://doi.org/10.1145/3392878

[26] IBM. 2019. TJBot IBM Research. https://www.research.ibm.com/tjbot/

[27] Deloitte Insights. 2019. Technology, Media, and Telecommunications Predictions. https://www2.deloitte.com/insights/us/en/industry/technology/technology-media-and-telecom-predictions.html

[28] Yangqing Jia, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross Girshick, Sergio Guadarrama, and Trevor Darrell. 2014. Caffe: Convolutional Architecture for Fast Feature Embedding. In *Proceedings of the 22Nd ACM International Conference on Multimedia* (Orlando, Florida, USA) *(MM '14)*. ACM, New York, NY, USA, 675–678. https://doi.org/10.1145/2647868.2654889

[29] Liu Jiang, Shixia Liu, and Changjian Chen. 2019. Recent research advances on interactive machine learning. *Journal of Visualization* 22, 2 (2019), 401–417. https://doi.org/10.1007/s12650-018-0531-1

[30] Minsuk Kahng, Pierre Y. Andrews, Aditya Kalro, and Duen Horng Chau. 2018. ActiVis: Visual Exploration of Industry-Scale Deep Neural Network Models. *IEEE Transactions on Visualization and Computer Graphics* 24 (2018), 88–97.

[31] Ashish Kapoor, Bongshin Lee, Desney Tan, and Eric Horvitz. 2010. Interactive Optimization for Steering Machine Classification. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Atlanta, Georgia, USA) *(CHI '10)*. Association for Computing Machinery, New York, NY, USA, 1343–1352. https://doi.org/10.1145/1753326.1753529

[32] Andrej Karpathy. 2016. Convnet JS Github. https://github.com/karpathy/convnetjs

[33] Keras. 2019. Keras Library. https://keras.io/

[34] J. Kim and C. Park. 2017. End-To-End Ego Lane Estimation Based on Sequential Transfer Learning for Self-Driving Cars. In *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. IEEE, Honolulu, HI, USA, 1194–1202. https://doi.org/10.1109/CVPRW.2017.158

[35] Todd Kulesza, Simone Stumpf, Margaret Burnett, and Irwin Kwan. 2012. Tell Me More? The Effects of Mental Model Soundness on Personalizing an Intelligent Agent. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Austin, Texas, USA) *(CHI '12)*. Association for Computing Machinery, New York, NY, USA, 1–10. https://doi.org/10.1145/2207676.2207678

[36] I. Lage, E. Chen, J. He, M. Narayanan, S. Gershman, B. Kim, and F. Doshi-Velez. 2018. An Evaluation of the Human-Interpretability of Explanation.

[37] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. 2015. Deep learning. *nature* 521, 7553 (2015), 436–444.

[38] Y. Liang, S. T. Monteiro, and E. S. Saber. 2016. Transfer learning for high resolution aerial image classification. In *2016 IEEE Applied Imagery Pattern Recognition Workshop (AIPR)*. IEEE, Washington, DC, USA, 1–8. https://doi.org/10.1109/AIPR.2016.8010600

[39] Laurens van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-SNE. *Journal of machine learning research* 9, Nov (2008), 2579–2605.

[40] S. J. Pan and Q. Yang. 2010. A Survey on Transfer Learning. *IEEE Transactions on Knowledge and Data Engineering* 22, 10 (2010), 1345–1359.

[41] Aditya Parameswaran. 2019. Enabling Data Science for the Majority. *Proc. VLDB Endow.* 12, 12 (Aug. 2019), 2309–2322. https://doi.org/10.14778/3352063.3352148

[42] Peter Pirolli and Stuart Card. 2005. The sensemaking process and leverage points for analyst technology as identified through cognitive task analysis. In *Proceedings of international conference on intelligence analysis*, Vol. 5. McLean, VA, USA, VA,USA, 2–4.

[43] Lorien Y. Pratt, Jack Mostow, and Candace A. Kamm. 1991. Direct Transfer of Learned Information among Neural Networks. In *Proceedings of the Ninth National Conference on Artificial Intelligence - Volume 2 (AAAI'91)*. AAAI Press, Anaheim, California, 584–589.

[44] Donghao Ren, Saleema Amershi, Bongshin Lee, Jina Suh, and Jason D Williams. 2017. Squares: Supporting interactive performance analysis for multiclass classifiers. *IEEE transactions on visualization and computer graphics* 23, 1 (2017), 61–70. https://doi.org/10.1109/TVCG.2016.2598828

[45] Sebastian Ruder, Matthew E. Peters, Swabha Swayamdipta, and Thomas Wolf. 2019. Transfer Learning in Natural Language Processing. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Tutorials*. Association for Computational Linguistics, Minneapolis, Minnesota, 15–18. https://doi.org/10.18653/v1/N19-5004

[46] Dominik Sacha, Michael Sedlmair, Leishi Zhang, John A Lee, Jaakko Peltonen, Daniel Weiskopf, Stephen C North, and Daniel A Keim. 2017. What you see is what you can change: Human-centered machine learning by interactive visualization. *Neurocomputing* 268 (2017), 164–175.

[47] Aécio Santos, Sonia Castelo, Cristian Felix, Jorge Piazentin Ono, Bowen Yu, Sungsoo Ray Hong, Cláudio T. Silva, Enrico Bertini, and Juliana Freire. 2019. Visus: An Interactive System for Automatic Machine Learning Model Building and Curation. In *Proceedings of the Workshop on Human-In-the-Loop Data Analytics* (Amsterdam, Netherlands) *(HILDA'19)*. Association for Computing Machinery, New York, NY, USA, Article 6, 7 pages. https://doi.org/10.1145/3328519.3329134

[48] Simone Stumpf, Vidya Rajaram, Lida Li, Weng-Keen Wong, Margaret Burnett, Thomas Dietterich, Erin Sullivan, and Jonathan Herlocker. 2009. Interacting meaningfully with machine learning systems: Three experiments. *International Journal of Human-Computer Studies* 67, 8 (2009), 639 – 662. https://doi.org/10.1016/j.ijhcs.2009.03.004

[49] Jina Suh, Soroush Ghorashi, Gonzalo Ramos, Nan-Chen Chen, Steven Drucker, Johan Verwey, and Patrice Simard. 2019. AnchorViz: Facilitating Semantic Data Exploration and Concept Discovery for Interactive Machine Learning. *ACM Trans. Interact. Intell. Syst.* 10, 1, Article 7 (Aug. 2019), 38 pages. https://doi.org/10.1145/3241379

[50] Justin Talbot, Bongshin Lee, Ashish Kapoor, and Desney S. Tan. 2009. EnsembleMatrix: Interactive Visualization to Support Machine Learning with Multiple Classifiers. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Boston, MA, USA) *(CHI '09)*. Association for Computing Machinery, New York, NY, USA, 1283–1292. https://doi.org/10.1145/1518701.1518895

[51] Lisa Torrey and Jude Shavlik. 2009. Transfer learning. *Handbook of research on machine learning applications and trends: algorithms, methods, and techniques* 1 (2009), 242–264.

[52] Malcolm Ware, Eibe Frank, Geoffrey Holmes, Mark Hall, and Ian H. Witten. 2002. Interactive Machine Learning: Letting Users Build Classifiers. *Int. J. Hum.-Comput. Stud.* 56, 3 (March 2002), 281–292.

[53] Karl Weiss, Taghi M Khoshgoftaar, and DingDing Wang. 2016. A survey of transfer learning. *Journal of Big data* 3, 1 (2016), 9. https://doi.org/10.1186/s40537-016-0043-6

[54] Marcus Winter and Phil Jackson. 2020. Flatpack ML: How to Support Designers in Creating a New Generation of Customizable Machine Learning Applications. In *Design, User Experience, and Usability. Design for Contemporary Interactive Environments*, Aaron Marcus and Elizabeth Rosenzweig (Eds.). Springer International Publishing, Cham, 175–193.

[55] Tongshuang Wu, Daniel S. Weld, and Jeffrey Heer. 2019. Local Decision Pitfalls in Interactive Machine Learning: An Investigation into Feature Selection in Sentiment Analysis. *ACM Trans. Comput.-Hum. Interact.* 26, 4, Article 24 (June 2019), 27 pages. https://doi.org/10.1145/3319616

[56] Qian Yang, Alex Scuito, John Zimmerman, Jodi Forlizzi, and Aaron Steinfeld. 2018. Investigating How Experienced UX Designers Effectively Work with Machine Learning. In *Proceedings of the 2018 Designing Interactive Systems Conference* (Hong Kong, China) *(DIS '18)*. Association for Computing Machinery, New York, NY, USA, 585–596. https://doi.org/10.1145/3196709.3196730

[57] Qian Yang, Jina Suh, Nan-Chen Chen, and Gonzalo Ramos. 2018. Grounding Interactive Machine Learning Tool Design in How Non-Experts Actually Build Models. In *Proceedings of the 2018 Designing Interactive Systems Conference* (Hong Kong, China) *(DIS '18)*. Association for Computing Machinery, New York, NY, USA, 573–584. https://doi.org/10.1145/3196709.3196729

[58] Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson. 2014. How Transferable Are Features in Deep Neural Networks?. In *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2* (Montreal, Canada) *(NIPS'14)*. MIT Press, Cambridge, MA, USA, 3320–3328.

[59] Matthew D. Zeiler and Rob Fergus. 2014. Visualizing and Understanding Convolutional Networks. In *Computer Vision – ECCV 2014*, David Fleet, Tomas Pajdla, Bernt Schiele, and Tinne Tuytelaars (Eds.). Springer International Publishing, Cham, 818–833.

[60] Fuzhen Zhuang, Zhiyuan Qi, Keyu Duan, Dongbo Xi, Yongchun Zhu, Hengshu Zhu, Hui Xiong, and Qing He. 2020. A comprehensive survey on transfer learning. *Proc. IEEE* 109, 1 (2020), 43–76.