$$f(x) \leq 5$$
$$x^2 - 4x + 5$$
$$x^2 - 4x \leq 0$$
$$h(B) + h(C) - h(B \cap C)$$
$$\sqrt[n]{a^3}$$
$$\sqrt[3]{a} = 3$$
$$\sqrt{5 + \sqrt{2}}$$
$$126 = 6xy$$
$$2x + 2y = 20$$
$$x = 5$$

FROM MACHINE LEARNING TO MACHINE TEACHING: THE IMPORTANCE OF UX

Martin Lindvall, Linköping University and Sectra AB

Jesper Molin, Sectra AB

Jonas Löwgren, Linköping University

Insights

- The realization of the potential of machine learning will require developments in the design of human-machine teaching environments.
- After initial ML components are trained, the UX designer can play a key role in creating man-machine systems that continuously improve through use.

From a human-centered perspective, one of the most important developments in the technical machine-learning (ML) domain is that learning algorithms can now improve their predictions when fed more training data. This means that processes and tools for generating training data, previously a matter mostly for one-off research projects, now have a large impact on the success of ML projects.

In many domains, the designers and engineers of machine-learning-based systems do not themselves hold the expertise required to create training data. In a typical project, the creation

and curation of data might require many man-months of effort and involve the collaboration of data scientists, machine-learning experts, and domain experts. Human-centered design methods can therefore play a key role in building systems that aid in the generation of training data.

A shift from tuning the performance of automated classifiers toward focusing on the *human-machine teaching* aspect of these interactions highlights the role of humans as teachers and their interaction with data as a key factor in building ML-based systems. The role of the designer and design activities

Figure 1. Overview of how our four designs (blue boxes) fit into a space defined by, on the one hand, how intelligent the AI component of the system already is (AI performance) and, on the other, the domain specificity of the interaction design effort required.

therefore grow in importance. One secret of ML is that human design decisions affect learning outcomes throughout the entire pipeline—from early inceptions about what use or role the ML component should play in a larger context to deciding what the system should predict. Since that which the system should predict is in turn strongly connected to the choice of which training data to gather and how to curate and label it, the UX practitioner can bring these issues to the fore at an early stage through careful attention to the design of aptly suited teaching environments.

To emphasize the role of UX practice in designing systems for users to generate training data, we will highlight some key ideas illustrated by examples drawn from our work within the domain of medical imaging. We will describe four interactive systems that have been created and used within digital pathology—diagnosing and reviewing digital gigapixel-size microscopic images of tissue samples such as biopsies and surgical specimens.

In the design of these tools, we have paid special attention to ensuring that manual, unassisted workflows are preserved and are compatible with the assisting tools. These examples together describe a typical two-step process we have used when designing new ML-based systems. First, when no a priori data exists, we need to

bootstrap a large enough dataset so that the algorithm used in the first version of the system performs sufficiently. Second, we need to ensure that the system can collect additional training data when it is deployed, by receiving user corrections. This will make the system self-sufficient on training data, enabling an incremental improvement of the AI performance. Put differently, we can view this as partitioning the design space according to AI performance and ensuring that the teaching interaction is suited to the machine's current level of "intelligence" (Figure 1). Our four annotated examples of this process are based on our own experience as UX designers active in the medical-imaging field. Two of the examples are prototypes, and two are finished products that we have either designed ourselves or followed closely.

EFFICIENT BOOTSTRAP TEACHING

An early step in the creation of an ML-based system, when no prior training data exists, is to create an initial dataset. For pathology images, this typically consists of drawing outlines over tissue regions and classifying them. Because it is a highly specialized domain, this usually means engaging pathologists. Since it is important to make efficient use of these individuals and their knowledge, it seems sensible to align the design of

the teaching environment with their experience.

Rapid interactive refinement. A well-known semi-automatic approach to assigning categories to visual regions is an *interactive segmentation tool*. The user of such tools typically uses a paintbrush-style interaction to assign areas to given categories (called *seeds*); after this is done, areas similar to the one marked are also assigned to the same category [1].

When we applied a human-centered design perspective to the construction of such a tool, we gained valuable insights. For our initial prototype (Figure 2), the interaction was experienced as a *trading of control between human and machine*, in which the human waits for the machine response after drawing an area. After a noticeable delay, sometimes a few seconds, the results are received and the human can make a correction, wait again, and repeat the process. Typically, the user would be both intrigued and annoyed by the automatic assignment of areas that were not specifically drawn over, sometimes resulting in long back-and-forth correction cycles without noticeable progress.

In a revised version, we aimed for *rapid fine-grained interaction*, in which spreading would be constructed as an incremental, collaborative effort between user and system, rather than being computed slowly but accurately in every coarse-grained step (Figure 3). The tool was changed so that the similarity threshold required for spreading increased with the distance from the original area. Additionally, we added precomputations so that results of user input typically arrive in less than 40ms. Combined, these changes allow working faster and more accurately, albeit while employing more mouse strokes. The more fine-grained interaction lets the user gradually develop a feel for the underlying algorithm and its limitations by observing many predictions over time.

Creating intrinsic incentive for teaching. Another approach to bootstrapping the initial training dataset is to design a useful manual tool that generates training data as a side effect. This approach is somewhat similar to ESP [2], a two-player guessing game that creates labeled

training data as a side effect of play. Inspired by this approach, it is possible to give clinicians manual tools to aid their daily decision making, and in that process use the clinicians' labels as the training data.

We have created one such tool to support pathologists in manual mitotic cell counting (Figure 4). In this diagnostic task, the pathologist should go through 10 fields of view in the highest magnification and count the number of cells undergoing cell division within that area. During this task, it can be challenging to keep track of the number of mitotic cells as well as the number of fields of view. In the tool, this task is supported by keeping track of the reviewed area when navigating the image. The user can click on detected mitotic cells, which are then stored. Upon completion, the cell density can be derived using the number of stored mitotic cells and the total tracked area. Even though the tool works by rather simple means, it still turns out to be very useful for the pathologist. The side effect is that every time a mitotic cell is clicked on, a training-data example is generated. Additionally, the tracked areas that were reviewed but not clicked on can be used as examples of non-mitotic cells. By deploying this tool into a delivered product, it will generate a bootstrapping dataset of mitotic cells that can be used to train an ML-based detection system.

DESIGNING FOR INCREMENTAL USER CORRECTIONS

Once ML systems are deployed, user corrections of the ML predictions can be used to generate additional training data. However, the designer needs to design specifically for this possibility. Our experience so far indicates that the most important factors for this type of design are to make sure that machine errors become apparent, that class labels are chosen in such a way that they are easy to interact with, and that there is an incentive in relation to the effort required.

This can be exemplified by an ML system used to quantify immunostains. Immunostaining is a technique used to chemically visualize protein expression in cells. A common protein used to quantify proliferation in tumor cells is KI-67. When the KI-67 immunostain is used, the nucleus becomes brown

Figure 2. The initial version of our interactive segmentation tool. The user draws a path (yellow) and waits for the response (light blue).

Figure 3. The revised segmentation tool. The user draws (shown in yellow), and results (light blue) update in real time. Each yellow circle represents a point in time where the user received updated results. The figure shows results for two points in time along the drawn path, as indicated with blue diamond markers.

Figure 4. A manual tool to help pathologists keep track of mitotic figures. This is used to generate training data for a future algorithm.

Association for
Computing Machinery

ACM Conference Proceedings Now Available via Print-on-Demand!

*Did you know that you can
now order many popular
ACM conference proceedings
via print-on-demand?*

Institutions, libraries and individuals can choose from more than 100 titles on a continually updated list through Amazon, Barnes & Noble, Baker & Taylor, Ingram and NACSCORP: CHI, KDD, Multimedia, SIGIR, SIGCOMM, SIGCSE, SIGMOD/PODS, and many more.

**For available titles and
ordering info, visit:
librarians.acm.org/pod**

Figure 5. An example of a symmetric input-output ML-system of a cell-counting system for KI-67 stainings.

if the cell is positive for this protein and appears blue from the background staining if it is not.

In the design of an ML pipeline, two apparent choices of class labels for this problem exist: pixel labels and nuclei labels placed on the center of the nuclei. If pixel labels are used, pixels belonging to positive and negative nuclei can be visualized to the user as an overlay on top of the original image, occluding the nuclei. If nuclei labels are used, the result can be visualized by placing glyphs on the center of each detected nucleus. Using glyphs makes it easier for the user to detect errors, since less ink is used to visualize the result and the original image becomes more visible. It also becomes easier to correct misplaced markers, since less precision is needed to click on markers than on pixels. The second approach was implemented as a product, shown in Figure 5.

For the user to be able to correct the results, in addition to actually seeing the underlying phenomena they must also be able to perform the validation with reasonable effort in relation to

perceived gain. For this design, we borrowed an heuristic from the light microscope and limited predictions to only 200 nuclei in the most prolific region.

This design also illustrates the seminal principle of direct manipulation [3]; the result is presented in an *input-output symmetric way* in which the user can directly manipulate the labeled data. By designing the system to allow for such direct manipulation, making the validation possible with reasonable effort and providing an *intrinsic incentive* in terms of the actual nuclei count, user corrections can be used to directly retrain and improve the underlying machine-learning model.

Our final example of an ML direct-manipulation interface is the patch gallery prototype shown in Figure 6, where the goal is to estimate the distribution of classes in an area. In this prototype, we generate a grid pattern over a user-selected area and extract a small image patch for each point in the grid. We then feed each

**We look forward to the point when our
user interfaces need to be redesigned
or augmented with interactions that are
adapted to ML components with much
higher performance.**

Figure 6. Patch gallery prototype, samples from the tissue are generated and classified by an ML algorithm into three classes, which the user can correct by drag 'n drop.

patch to a trained ML component that classifies the patches into different categories, which are then shown in a sorted gallery. Each defined class in the trained model is shown as a patch in the same gallery, and the user can then 1) click on a patch to see it in the main view to get a sense of its context in the tissue, and 2) change a label by either dragging the patch to the correct category or by clicking on the button or the corresponding shortcut key. Note that, similar to the previous example, the output of the classifier is limited to producing predictions not for the entire area, but only for a representative systematic sampling of that area, which makes the validation and correction effort tractable.

Like the mitotic counter in the previous section, both systems share the property that the generated parameter that the clinician wants to assess can be derived from manual input only. If the nuclei-detection algorithm failed to detect any nuclei, the user could still manually click on all the nuclei to calculate the KI-67 index. However, the amount of clicking would likely overwhelm the user. These user-correction systems do not strictly need an ML component, but practical usability requires automated support with a certain level of prediction accuracy.

Another crucial factor when designing this type of system is that the

user-correction accuracy needs to be higher than that of the ML component alone, in order for the generated training data to add value when retraining the ML component.

DISCUSSION

In the design of these tools, we have paid special attention to ensuring that manual, unassisted workflows are preserved and compatible with the assisting tools. When possible, training-data collection has been designed to become an integrated part of clinicians' daily diagnostic practice. Furthermore, as the predictive performance of our initial ML components improve using the clinicians' corrections as training data, we look forward to the point when our user interfaces need to be redesigned or augmented with interactions that are adapted to ML components with much higher performance. It is our ambition to design the new interfaces so that the user can provide corrections and simultaneously teach and verify results at increasingly higher levels, forming a verification staircase [4], as opposed to a steep cliff where the user has to validate all or nothing.

Looking forward, as the technology comes of age, we hope this human-centered *teaching* aspect will continue to spark novel designs as interactions expand from dyadic teacher-learners to networks of machine intelligences

and groups of human specialists co-evolving, teaching, and learning in daily practice.

Observing ML-based product development from the view of training-data generation, we have shown how decisions made by the UX designer have an enormous impact on project success. Each step of training-data generation needs to get the motivations right so that users are willing and able to provide corrections. The choice of what the training dataset should consist of and thus what the ML component should predict is tightly connected to how the user interface should look, behave, and be interacted with. Hence, in the creation of human-centered machine-learning systems, the UX designer plays a key role from the start and throughout.

ENDNOTES

1. McGuinness, K. and O'Connor, N.E. A comparative evaluation of interactive segmentation algorithms. *Pattern Recognition* 43, 2 (2010), 434–444.
2. von Ahn, L. and Dabbish, L.. Labeling images with a computer game. *Proc. of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, New York, 2004, 319–326.
3. Shneiderman, B. The future of interactive systems and the emergence of direct manipulation. *Behaviour & Information Technology* 1, 3 (1982), 237–256.
4. Molin, J., Woźniak, P.W., Lundström, C., Treanor, D., and Fjeld, M. Understanding design for automated image analysis in digital pathology. *Proc. of the 9th Nordic Conference on Human-Computer Interaction*. ACM, New York, 2016, Article 58.

👤 **Martin Lindvall** is an industrial Ph.D. student exploring interactions using machine learning as a design material. Currently part of the Wallenberg AI, Autonomous Systems and Software Program (WASP), his background includes an M.Sc. in cognitive science and 10 years of experience designing and developing medical information systems at Sectra.
→ martin.lindvall@sectra.com

👤 **Jesper Molin** is research scientist and UX designer at Sectra, exploring and designing ML-based tools used within clinical routine pathology. His background includes an M.Sc. in applied physics and electrical engineering and a Ph.D. in human-computer interaction from Chalmers University of Technology.
→ jesper.molin@sectra.com

👤 **Jonas Löwgren** is professor of interaction and information design at Linköping University, Sweden. His expertise includes collaborative media, interactive visualization, and the design theory of digital materials.
→ jonas.lowgren@liu.se