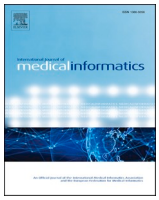
International Journal of Medical Informatics 183 (2024) 105337

Contents lists available at ScienceDirect

International Journal of Medical Informatics

journal homepage: www.elsevier.com/locate/ijmedinf



Application of a human-centered design for embedded machine learning model to develop data labeling software with nurses: Human-to-Artificial Intelligence (H2AI)

Naomi A. Kaduwela [a], Susan Horner [b], Priyansh Dadar [a], Renee C.B. Manworren [b,c,*]

[a] KaviGlobal, 1250 Grove St, Suite 300, Barrington, IL, USA
[b] Ann & Robert H. Lurie Children's Hospital of Chicago, 255 E. Chicago Ave, Box 101, Chicago, IL, USA
[c] Northwestern University Feinberg School of Medicine, Department of Pediatrics, 255 E. Chicago Ave, Chicago, IL, USA

ARTICLE INFO

Keywords:
Clinical decision support software
Data labeling
Human-centered Design for Embedded Machine Learning Solutions Machine Learning
Machine learning models

ABSTRACT

Background: Nurses are essential for assessing and managing acute pain in hospitalized patients, especially those who are unable to self-report pain. Given their role and subject matter expertise (SME), nurses are also essential for the design and development of a supervised machine learning (ML) model for pain detection and clinical decision support software (CDSS) in a pain recognition automated monitoring system (PRAMS). Our first step for developing PRAMS with nurses was to create SME-friendly data labeling software.
Purpose: To develop an intuitive and efficient data labeling software solution, Human-to-Artificial Intelligence (H2AI).
Method: The Human-centered Design for Embedded Machine Learning Solutions (HCDe-MLS) model was used to engage nurses. In this paper, HCDe-MLS will be explained using H2AI and PRAMS as illustrative cases.
Findings: Using HCDe-MLS, H2AI was developed and facilitated labeling of 139 videos (mean = 29.83 min) with 3189 images labeled (mean = 75 s) by 6 nurses. OpenCV was used for video-to-image pre-processing; and MobileFaceNet was used for default landmark placement on images. H2AI randomly assigned videos to nurses for data labeling, tracked labelers' inter-rater reliability, and stored labeled data to train ML models.
Conclusions: Nurses' engagement in CDSS development was critical for ensuring the end-product addressed nurses' priorities, reflected nurses' cognitive and decision-making processes, and garnered nurses' trust for technology adoption.

1. Introduction

Within healthcare organizations, clinical decision support software (CDSS) operates as integral components of complex sociotechnical systems profoundly influenced by social (people and environment) and technical (technology and tasks) subsystems [1]. Understanding this dual influence is essential for effective CDSS design. The Human-Centered Design for Embedded Machine Learning Solutions (HCDe-MLS) model provides a systematic approach (Fig. 1) to developing innovative CDSS solutions for complex healthcare challenges. HCDe-MLS uses automation, human factors, machine learning (ML), and systems engineering to advance human–machine interactions and human-systems integration [2]. Although HCDe-MLS first focuses on users and stakeholders, it is a holistic systems approach because human-centered solutions designed at the micro-system level to assist healthcare professionals perform specific tasks, also influence the larger *meso*-systems (healthcare teams and information technology integration) and macro-systems (for example, healthcare organizations, healthcare standards and policies, and healthcare financing) [3]. In other words, CDSS designed to support patient monitoring tasks influence physical, cognitive, behavioral, and organizational processes, as well as outcomes of healthcare systems over time.

Abbreviations: CDSS, clinical decision support software; H2AI, Human-to-Artificial Intelligence; HCDe-MLS, Human Centered Design for Embedded Machine Learning; IRR, inter-rater reliability; NCSF, Neonatal Facial Coding System; NICU, neonatal intensive care unit; PRAMS, Pain Recognition Automated Monitoring System; SME, subject matter expert; sQuaRE, System and Software Quality Requirements and Evaluation; VAS, visual analog scale.

* Corresponding author at: Ann & Robert H. Lurie Children's Hospital of Chicago, 225 E. Chicago Avenue, Box 101, Chicago, IL 60611-2991, USA.
E-mail addresses: Naomi.kaduwela@kaviglobal.com (N.A. Kaduwela), SHorner@luriechildrens.org (S. Horner), Priyansh.Dadar@kaviglobal.com (P. Dadar), Renee.Manworren@northwestern.edu, RCBManworren@luriechildrens.org (R.C.B. Manworren).

https://doi.org/10.1016/j.ijmedinf.2023.105337
Received 14 August 2023; Received in revised form 16 December 2023; Accepted 30 December 2023
Available online 6 January 2024

The three core characteristics of human-centered design are understanding users, stakeholder engagement, and a systems approach [3]. Understanding nurses and nursing workflows are essential for designing CDSS that meets nurses' needs. Engaging nurses throughout the design and development lifecycle ensures that the end-product addresses nurses' priorities and solves pragmatic problems [4]. By prioritizing nurses' needs and human–machine interactions, CDSS can be designed to efficiently accomplish tasks, while ensuring nurses' experiences are intuitive and meaningful. Ideally, CDSS reflects nurses' actual cognitive and decision-making processes [5].

1.1. HCDe-MLS and software quality

According to the System and Software Quality Requirements and Evaluation (sQuaRE) standards, software evaluation should include five quality-in-use and eight product quality characteristics [6,7]. Usability is the sQuaRE characteristic most often evaluated [8]; however, individual factors are most important for influencing perceptions of software quality [9]. The HCDe-MLS model ensures user's perceptions of software quality are influenced by individual, technological, and organizational factors.

Trust, or the certainty that a system will not fail, is a critical driver of technology usage behaviors and is essential for user adoption of technology [10]. Product-related factors, such as perceived usefulness, helpfulness, functionality, reliability, and ease of use, as well as security/service-related and social factors influence trust and CDSS adoption. Design elements of the user interface are an important predictor of users' trustworthiness in CDSS [11]. Thus, the HCDe-MLS model of including users in software development is critical for influencing nurses' involvement, training, knowledge, competency, resistance to change, and overall perceptions of CDSS quality.

1.2. Purpose

The purpose of this study was to develop an intuitive and efficient data labeling software solution, Human-to-Artificial Intelligence (H2AI). This paper describes the HCDe-MLS model and its use to develop the H2AI software solution. H2AI facilitates data labeling by subject matter experts (SMEs), enables tracking of data labeling progress, and allows monitoring of data labeling quality with inter-rater reliability (IRR) dashboards. We leveraged the HCDe-MLS model to maximize neonatal intensive care (NICU) nurses' engagement, experience, and productivity to develop H2AI, an intuitive ML model data labeling software solution. In this case, H2AI was developed and used to train a ML neonatal pain classification model, a Pain Recognition Automated Monitoring System (PRAMS).

2. Methods: HCDe-MLS model

The HCDe-MLS model combines human thinking and ML lifecycles (Fig. 1).

2.1. Human thinking lifecycle

The human thinking lifecycle seeks to understand human needs (Table 1). Human thinking requires imagination, logic, and systematic reasoning to artfully create user-focused outcomes [12]. The ML lifecycle seeks to find patterns in existing data, apply these patterns to new data, and embed ML in solutions. ML lifecycle follows the standard agile and iterative software development stages [13].

2.1.1. Empathize

In the human thinking lifecycle, **empathizing** focuses on learning about target users. The purpose of empathizing is to set aside assumptions and instead gain insights into users' actual physical, cognitive, and emotional needs to complete tasks [3,14]. However, Boy [2] clarifies

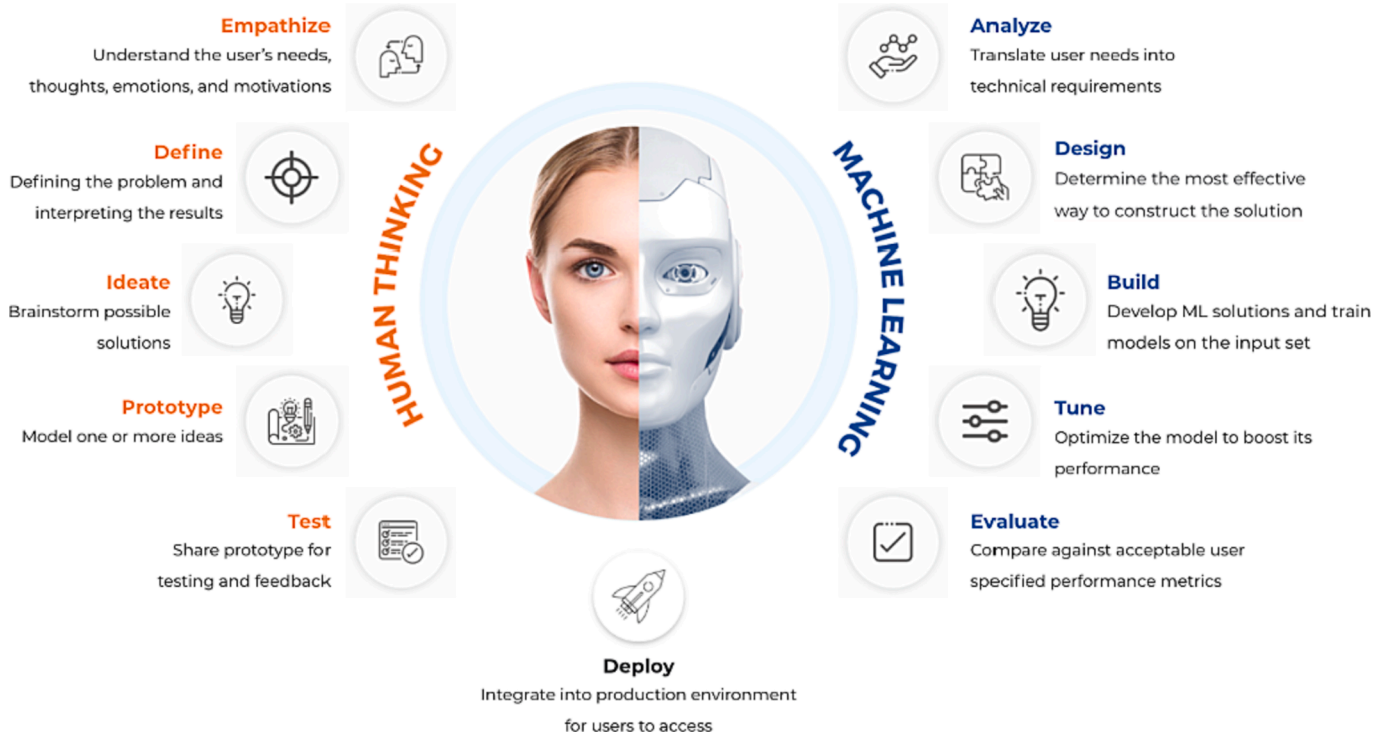


Fig. 1. Human-Centered Design for Embedded Machine Learning Solutions (HCDe-MLS): Human thinking (*Left side of diagram*) from Empathize to Test, and then the Machine Learning lifecycle (*Right side of diagram*) from Analyze to Evaluate. When both are complete, the solution is deployed (*Bottom of diagram*). Reprinted with permission from ©2019 Kavi Global.

Table 1
Human Thinking Lifecycle Stages and Respective Toolbox.

Lifecycle stages	Empathize	Define	Ideate	Prototype	Test
Description	• Collect information • Gather insights	• Synthesize insights • Microtheory of user problem and needs • Validate with users	• Generate ideas for possible solutions to defined problems and needs	• Build low- and high-fidelity tactile representations of solutions	• Generate performance data • Gather feedback from users and stakeholders
Tools	• Interviews • Focus groups • Surveys • Storytelling • Generative technique	• Empathy mapping • User persona • Journey mapping	• Brainstorming • Mind-mapping • Affinity diagram • Co-creation	• Feature v1/v2 sketches • Visual prototypes	• Feedback grid

that tasks, such as pain assessment, are technology-centered prescriptions to humans, and activities are what humans really do. Empathizing requires listening, engaging, observing, and understanding users to gain insight into human activities [12,15].

2.1.2. Define

Empathizing is followed by ***defining*** the problem and user requirements. Through qualitative research methods, insights are synthesized, and user personas are developed. Personas are detailed descriptions of target users developed from highly specific data about real people [16]. The aim of using personas is to create the users' point of view, reframe the problem, and effectively focus design efforts on users' needs and preferences. Defining the problem brings clarity to ensure the solution solves the true problem in the best way [12].

2.1.3. Ideate

Engaging users in brainstorming. Mind-mapping or co-creation sessions initiates ***ideation*** [3]. The aim of the ideate stage is to channel empathy, familiarity, creativity, and collective situational awareness to address the shared purpose by developing a broad range of possible solutions that are unbounded by the limitations and status quo of the current state [2]. Then, all the possibilities must be evaluated against the constraints of resources and context to prioritize and finalize the most feasible solution [12,15].

2.1.4. Prototype

The best idea is then built as a ***prototype***. Prototypes may range from low-fidelity sketches to high-fidelity working artifacts [3,4]. Effective prototypes communicate concepts and test ideas through iterative feedback from users and stakeholders. Prototypes are important for implementing possibilities and for maintaining a solution-building approach [12].

2.1.5. Test

The final stage of the human thinking lifecycle involves ***testing*** and refining of the software created in the prototype stage [12]. Essential components of testing include representative users, stakeholders, tasks, and environments. Qualitative and quantitative methods are used to identify problems, capture recommendations for improvement, and statistically support qualitative concerns [15]. Tangible metrics should be developed with users and stakeholders to improve the assessment of complex system interoperability [2].

2.2. Machine learning lifecycle

The first stage of the ML lifecycle is to **analyze** user needs and translate needs into requirements.

2.2.1. Analyze

The analyze stage advances tasks to activity and complexity analysis [2]. The scope and boundaries of software, the functional and technical requirements, the nonfunctional requirements, data sources, data collection, and integration in a format that can later be consumed by the ML model, application, and user are defined [17]. Functional requirements, including inputs, calculations, and processes, are then translated into technical requirements of how the software performs its actions. Nonfunctional requirements are the look and feel of the product, the user interface and experience. A core focus in this stage is data management, including obtaining data essential for the process of training, testing, and validating the ML model [13]. Data collection requires gathering data samples of real-world system, process, or phenomenon for which the ML model is being built. The data collected may be heterogeneous because of various disparate sources; thus, preprocessing the data to ensure consistency is inevitable. When data samples are unavailable or their collection is too costly, time consuming, unethical or dangerous, augmentation methods are used to add these critical data samples to collected data sets [18].

2.2.2. Design

The ***design*** stage is the most creative in the ML lifecycle. Here, focus transitions from the problem to the solution to design optimal solution architecture leveraging technologies to solve problems efficiently and effectively. The goal is to transform the requirement specifications into structure. Creativity, system thinking, risk taking, agile approaches, and knowledge of human systems integration architecture is required [2]. An outline of the solution is generated, including the technical approach, solution architecture, ML models, evaluation metrics, capability of the team, project constraints, risks, timeline, and budget. Solution features are prioritized based on complexity, speed to value, and cost to determine the optimal minimal viable product. Buy versus build decisions are made for technology and components, as well as leveraging accelerators, for example, existing pre-trained models like Convolutional Architecture for Fast Feature Embedding (Caffe)-based convolutional neural network (CNN). CNN is a feed-forward neural network that uses filters to effectively extract information from images. Hsu et al. [19] introduced a CNN-based model to detect 68 facial landmarks on facial images.

Visual appeal and usability can override trust in information quality; however, accuracy is one factor that stimulates reflection and motivation for information quality [11]. CDSS performance requires access to data sets and multimodal healthcare data that can be assessed cognitively and longitudinally to make dynamic predictions and reflect timing of clinical decision making [5]. Predictive models must know the dimensionality of the data, for example, the strong predictive value of International Classification of Disease codes (ICD-10-CM) and Diagnosis-Related Groups (DRGs) and a priori interactions of clinical data. When large de-identified data sets are used to train predictive ML models, historical mistakes in datasets, known as "historical decision bias," are carried forward in model [5,13]. ML model performance improves when temporal changes and trends of repeated measurements are considered. For dynamic predictions of clinical outcomes, models can be trained "on-the-fly" [5]. Unfortunately, on-the-fly training of Bayesian

models results in reduced model performance, and on-the-fly training of computationally expensive complex algorithms (e.g., support vector machines [SVM] and CNN) result in slow responses and limited CDSS utility. Thus, to manage performance, most clinical decision support models are trained in nightly or weekly batches and only scoring of a new patient record is done on-the-fly in real time.

2.2.3. Build

Model building is the process of implementing ML models (e.g., logistic regression, SVM, random forest, and deep learning models like CNN) to solve the identified problem. Model building follows data preprocessing, and encompasses feature engineering, splitting of data into training data and test data, and running various models on the training set. ML models are broadly classified as supervised and unsupervised; the learning process is defined as classification or regression [20]. Supervised learning algorithms learn to map inputs to outputs based on labeled input–output training data pairs. Supervised learning may define outputs by classification (resulting in a finite set of output categories) or by regression (defining the probability of the output based on the input). Model selection is based on the type of problem, volume, and availability of training data, as well as the need for model transparency and explain-ability [13,18,21].

2.2.4. Tune

Some ML models have hyperparameters, which are used to control the learning process and can be iteratively ***tuned*** to optimize model performance and results. Tuning is the stage of improving ML model performance by choosing and optimizing the hyperparameters of the training algorithm to control for overfitting, underfitting, and model complexity [13,18,21]. ML models lack design specifications; instead, algorithms are developed by learning parameters from mathematically derived data. With models that do not require hyperparameter tuning, i.e., pre-trained models like Caffe-based CNN and MobileFaceNet, the tuning stage is unnecessary. However, in general, model performance can be improved by iterating on the features fed into the model.

2.2.5. Evaluate

The performance of the chosen model is then ***evaluated*** against the original use requirements and acceptance criteria on previously unseen test data [13]. Model evaluation demonstrates the robustness and generalizability of the model and enables comparison to other existing methods. Performance metrics should be quantifiable and reflect data characteristics and the CDSS [13,22]. For supervised models, performance metrics typically include accuracy, precision, recall, and speed. Especially in the healthcare context, it is important to evaluate tradeoffs between types of error (i.e., false positives and false negatives) to ensure patients are not misclassified or incorrectly treated. Evaluation metrics must also consider human interpretation of what the algorithm does and means [22].

2.2.6. Deploy

The last stage of the HCDe-MLS model is ***deployment*** to the production environment. Deployment refers to configuring the CDSS for integration with other applications to serve as designed at scale. Built-in mechanisms to integrate feedback and support CDSS may be required [13]. Human-machine interfaces must enhance operator automation-related situational awareness [23]. Failing to attend to the knowledge, expertise, and training to optimize human–machine interactions results in automation errors. In addition, deviation from test data to operation data must be monitored to identify covariance shift or concept drift [13,21]. Therefore, it is essential that, in safety–critical systems like healthcare, any deployed model is transparent, explainable, interpretable, and continuously monitored to meet clinical decision support needs [21].

3. Stakeholders and setting

After receiving approval from the Institutional Review Board, study #2021-4348, small focus groups were conducted by video calls for 1 h each week from February to May in 2021 to ***empathize*** and identify user needs. Stakeholders included NICU nurses (n = 6), nurse scientists (n = 2) with expertise in neonatal development and pain management, human-centered design specialists, architects, data scientists, and product managers. Nurses had a mean of 18.7 years of NICU nursing experience (ranging from 5 to 42 years) and worked in a 64-bed level IV NICU, part of a 364-bed, free-standing, university-affiliated, not-for-profit urban children's hospital in Illinois that cares for neonates with complex medical needs.

4. Results: H2AI development case

Our cross-functional team identified an opportunity to train a variety of ML models by labeling data. Models could then be compared against the nurses' benchmark to gain clinical trust and encourage CDSS adoption. We developed user personas to ***define*** user tasks and needs through thematic analysis by the human-centered design specialists and verification from all focus group members (Table 2). These personas provided real-life context to reframe the problem and focus design efforts toward efficiently leveraging nurses' expertise for data labeling; and eventually, development of an effective PRAMS.

Then, our cross-functional team identified novel ***ideas*** and disruptive innovations to optimize user workflows, maximize productivity, and minimize user burden. Our resulting mind map (Fig. 2) illustrates the key H2AI product features identified.

4.1. Data labeling tasks

Six Data Labeling Tasks were defined based on clinical neonatal pain assessment standards [24], a review of the literature [25,26], and ML modeling needs [6,7,13,18,21]. First, nurses used the Neonatal Facial Coding System (NFCS) to label each video frame. NFCS is a valid and reliable objective measure of pain [24,27,28]. Second, nurses rated their perception of pain intensity on a Visual Analog Scale (VAS) of 0–100, with 0 indicating no pain and 100 indicating the worst possible pain. Third, nurses identified and labeled facial landmarks to help the computer vision model identify facial action units from movement of facial features. Fourth, nurses identified occlusions, where neonates' hands or blankets obstruct facial landmarks. Fifth, nurses classified pain by frame image, and sixth, at the video level.

4.2. User workflows

Four User Workflows were developed.

4.2.1. Practice workflow

The purpose of the *Practice* workflow was to educate nurses in the tasks and features of the application. Since users had identified that they would need to access the application from a variety of computers, the *Practice* workflow was also used to test their equipment.

4.2.2. Training workflow

The *Training* workflow was created to ensure consistency of labeling among nurses. A nurse scientist labeled five random frames in parallel with each nurse, then the two met to reconcile any labeling differences. If agreement thresholds were reached before meeting, the nurse was "passed" on to the *Labeling* workflow. If thresholds were not attained, parallel labeling continued in repeated sets of five additional frames until the thresholds were reached.

4.2.3. Labeling and review workflows

The *Labeling* workflow was identical to the *Training* workflow, except

Table 2
User Personas.

We are…	We are trying to…	But…	Because…	We need to create solutions that…
Nurse Scientists	Automate pain classification in neonates	We need a ML model that healthcare professionals will trust	In healthcare, risk from false positives and false negatives is high	Innovate and involve direct care nurses in the rigorous development of a continuous pain monitoring system for vulnerable neonates
Architects & Data Scientists	Build a supervised ML model to automate pain classification	We need nurse-labeled data and a method to collect the data labels to train the ML model	We want the model to be trustworthy, and therefore comparable to SME benchmarks and validated methods of pain classification	Inspire and partner with healthcare professionals to develop an efficient solution for ML modeling
NICU nurses	Label neonatal facial landmarks and facial action data to train the ML model	Variability in assessments among nurses are normal; documenting each assessment and decision is time-consuming	There are so many landmark points and NCSF pain classification results, pain intensity, and overall pain classification to capture for each frame	Empower nurses to engage in designing the labeling system and the development of a clinical decision support solution to provide better care for my patients

ML, machine learning; NCSF, Neonatal Facial Coding System; NICU, neonatal intensive care unit.; SME, subject matter expert.

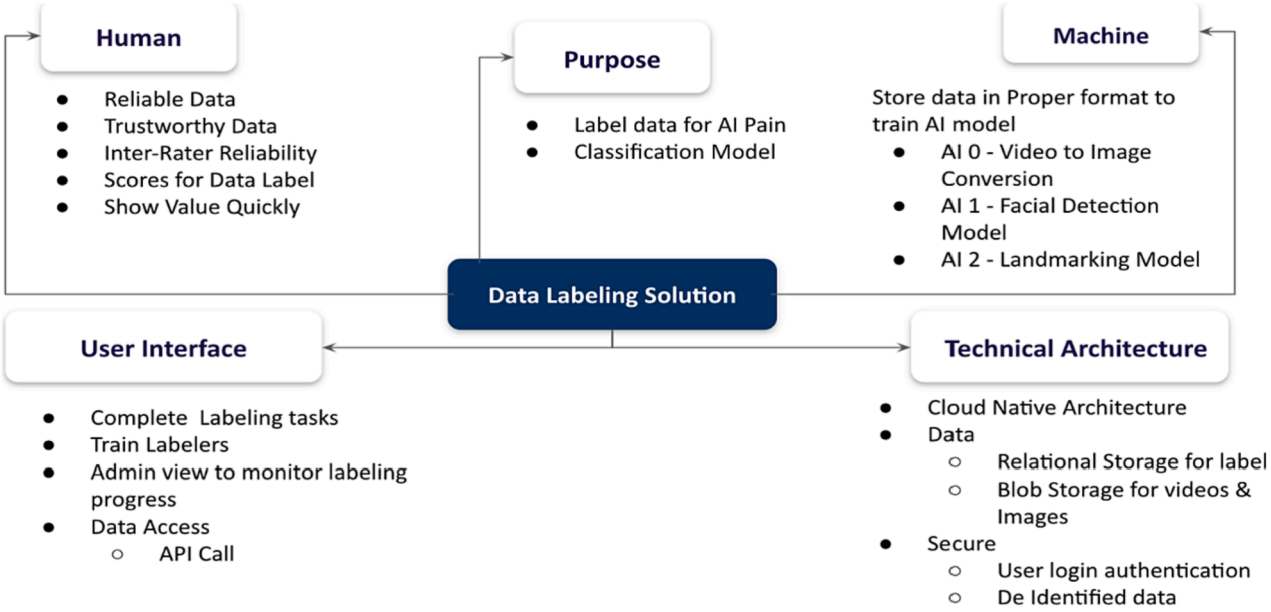


Fig. 2. Mind map of key features for embedded ML solutions development. AI, artificial intelligence; API, application programming interface.

the generated data labels were stored for later use to train the ML pain classification model. To ensure nurses consistently met IRR thresholds throughout the labeling of thousands of video frames, the nurse scientists were randomly assigned to label up to 10 % of the videos each nurse labeled. Given the volume of frames to be labeled, the *Review* workflow was created to allow real-time monitoring of data labeling progress (i.e., how many frames/videos were labeled, how long each task takes, and IRR for each nurse).

4.3. H2AI prototype

When ready to create a ***prototype,*** the data scientist first conducted a buy versus build comparison to determine if the data labeling capabilities already existed in the market. Image annotation solutions already existed. Common features were pixel identification, bounding box, region detection, text tagging/object, however, none provided the ability to upload data based on human interactions with video images. Mid- and high-fidelity prototypes were then built (Fig. 3) and ***tested*** by nurses. Our feedback grid both itemizes improvement opportunities and positive feedback (Table 3).

4.4. H2AI machine learning lifecycle

Feedback was ***analyzed*** and mapped to the product backlog to optimize functionality, user experience, and productivity. Data security agreements required user authentication and secondary verification. This greatly influenced the architectural ***design*** and ML modeling approach. This solution was funded on a time-limited grant which required a software solution be in production in three months. User requirements were translated into technical requirements, mapped to the appropriate technology and ML model solution (Table 4), and then consolidated into a single cohesive solution. The solution architecture (Fig. 4) encompasses a holistic software solution from front-end user interface to the embedded ML models output, back-end data storage, and service calls to pass data between the front and back ends.

4.4.1. H2AI build

To build the ML model, neonatal pain and no pain video images from the iCOPEvid Neonatal Pain Video Database was obtained with permission and used for this study [25]. Videos and images needed to be labeled by nurses in the data labeling solution. H2AI utilizes pre-trained models that are optimized to extract facial features from video frames with high efficiency and capture labels at the lowest level of granularity. Intel's open-source framework, OpenCV, has a built-in Face Detector that is reliable in 90–95 % of clear, forward- and camera-facing human photos ([29], Open CV). OpenCV was selected to convert video to images, crop the face, and put the bounding box on the face to position facial landmarks within the acceptable level of confidence (Fig. 5). The default OpenCV model cropped the outline of the face, especially by the ears and chin; thus, additional padding of 20 pixels were added before

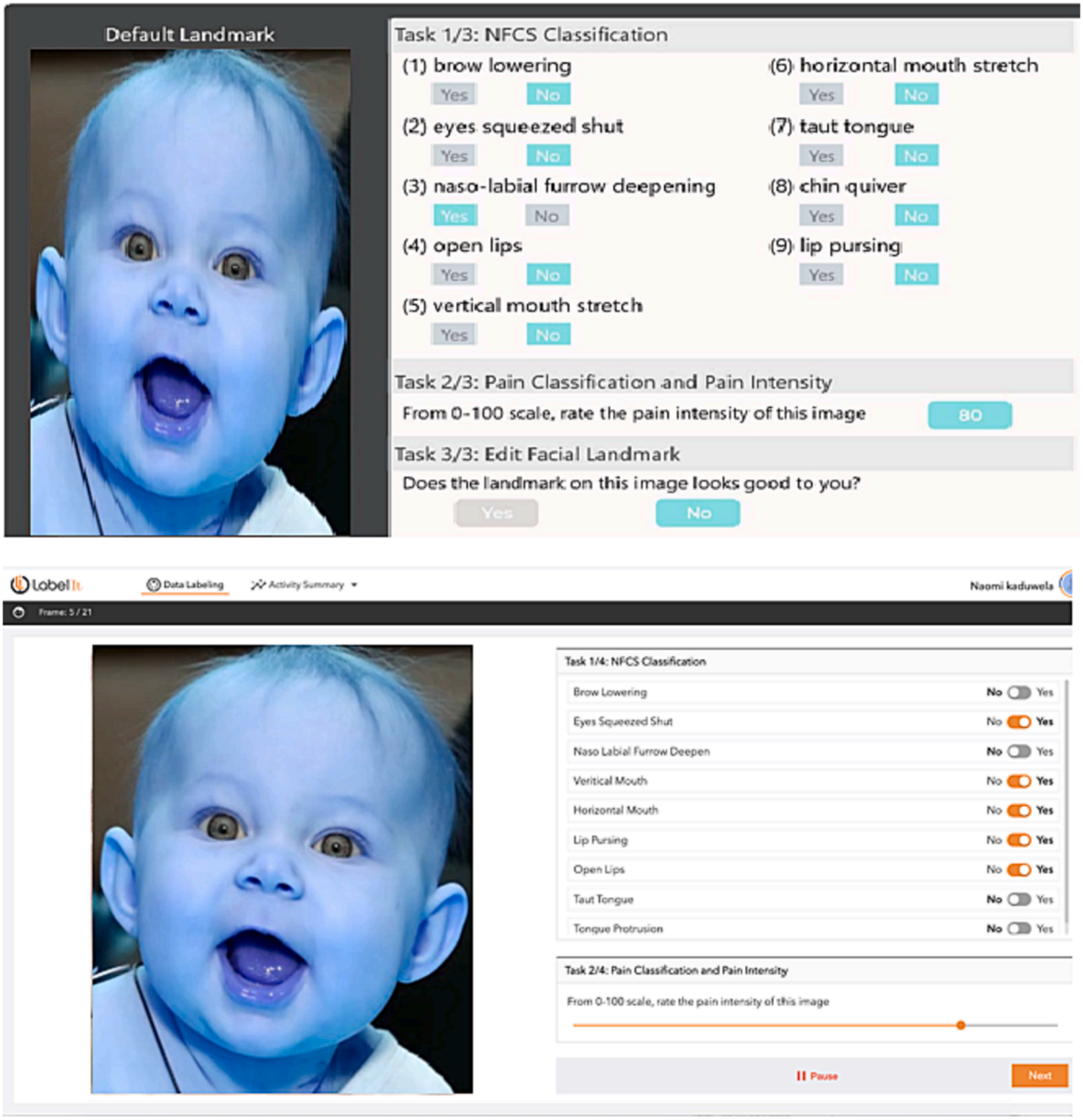


Fig. 3. Mid-fidelity App (top) and high-fidelity App in production (bottom).

cropping the image. This ensured that all facial features were available for landmarks that might otherwise be lost.

4.4.2. H2AI landmark model comparison

Two pre-trained facial landmark models were implemented, and precision of their respective landmark placements were compared. First, a Caffe-based CNN model was implemented. Caffe is a deep learning framework that defines a net layer-by-layer in its own model schema. The network defines the model in a bottom-to-top approach from input data to loss. The model was composed of 24 layers: 8 convolutional layers, 4 pooling layers, 2 dense layers, 9 batch-normalization layers, and 1 flatten layer. Using Keras Functional Application Programming Interface (API), the pre-processed frames of images were fed into the model.

The second model, MobileFaceNet, uses a more streamlined architecture with depthwise separable convolutions [30]. Chen et al. [31] developed MobileFaceNet, using ArcFace [30] loss to achieve $> 99.5\ \%$ accuracy for the face detection task on the Labeled Faces in the Wild Home (LFW) dataset [32]. MobileFaceNet is also effective as a general facial feature extractor [33]. MobileFaceNet is specifically designed for the face recognition task by replacing the global average pooling layer with a global depthwise convolution (GDConv) layer, which enhances the discriminative ability of the model. The first layer of each sequence uses a stride s, and all other layers use stride $= 1$ to preserve the same output feature map size as the original layer. All spatial convolutions in the bottlenecks use 3×3 kernels. The expansion factor t is always applied to the input size and GDConv7$\times$7 denotes GDConv of 7×7 kernels. A downsampling strategy is used at the beginning of the network, and a linear 1×1 convolution layer follows a linear global depth-wise convolution layer as the feature output layer. During training, batch normalization is used, and batch normalization folding is applied before deployment.

Table 3
Feedback Grid.

Feedback	Nurses' current human–machine interaction paradigm	Solution Feature Enhancement
We need the data labeling to be more efficient	Medical records allow nurses to copy forward	Copied forward previously selected results from tasks across frames
Increase size of dots on user interface • Facial landmarking dots are too small to see, select, and move • Pain VAS slider is too small	Nurses use a variety of computer brands, monitor sizes, trackpads, mouse, etc.	• Increased (Task 2 & 3) dot size • Increased (Task 4) pain intensity slider granularity
Need fine-grain control to move facial landmark dot(s)		Enabled single select, multi-select, rotation, space expansion, and space contraction of a group of landmarks dots at once
Limit risk of user pain intensity score bias	• Numeric pain scales have inherent bias • VAS is a valid pain intensity measure with more rigor	Hid numbers on pain intensity slider
Need practice and training workflows for training nurse data labelers and ensuring data quality	Goal is to maximize IRR	Created workflows: practice, training, and labeling, with user-specified IRR thresholds that need to be passed in training before entering labeling workflows
Cannot identify chin quivering due to use of still image	May negatively influence IRR	Removed chin quiver from NFCS
Need a way to monitor nurses' progress in training and labeling	• Variable schedules due to patient demands • Encourage and reward efforts • Track paid time • Need to identify data drift	A Power BI dashboard was embedded into the user interface to summarize the progress of nurses
Need better default landmark placement at the start of each labeling task	To improve efficiency by having to move fewer landmark points into place	Updated pre-trained facial landmarking default placement AI model from OpenCV to MobileFaceNet

AI, artificial intelligence; BI, business intelligence; IRR, inter-rater reliability; NFCS, Neonatal Facial Coding System; VAS, visual analog scale.

Since Caffe-based CNN and MobileFaceNet are pre-trained models, we did not **tune** hyperparameters for landmark detection. However, we adjusted the size and color tone of input images to achieve the best results. Models were then compared and **evaluated** using visual inspection across several images, including challenging images with occlusions. As seen in Fig. 6, the Caffe-based CNN model lacked precision; and MobileFaceNet better captured the outline of the upper lip (versus the tongue), nose, and eyes. Users agreed that MobileFaceNet was the better solution for default facial landmarking and was more robust at handling occlusions and blurry images from movement. Therefore, we integrated and ***deployed*** this pre-trained model into the production environment; and nurses, who had met IRR thresholds, then began data labeling workflows.

4.4.3. H2AI efficiency evaluation

Using HCDe-MLS, H2AI was developed and facilitated labeling of 139 videos with 3189 images labeled by 6 nurses. Nurses began labeling data after meeting IRR thresholds of 88 % agreement were attained on NFCS items and binary pain classification, and when agreement on pain intensity scores were ±10 points across 5 random test frames [34]. NFCS labeling took nurses a mean of 12.23 s per image and 4.67 min per video.

Table 4
Technical Requirements and Appropriate Technology and ML Model Solution.

User Requirement	Technical Requirement	Solution
User needs to provide data labels for six tasks on each image frame from each video.	Video data needs to be pre-processed into images and made available in the data labeling solution for users to label.	Video data can be stored in a blob format. OpenCV is the most popular image processing library to capture images from videos and detect faces.
User wants landmarks to be as precise as possible for optimal landmarking efficiency.		Pre-trained landmark models can be run in the back end to place default landmarks as close as possible to outline facial features. There are several options: Caffe Model and MobileFaceNet can be compared.
	Image and default landmark positions need to be made available in the user interface.	Image path and default landmark positions can be sent via Restful API call via JSON file to the user interface, which can then display the coordinates on the UI, over the image file.
Users want to automate pain detection using the validated NFCS pain scale measures.	Labeled data from the users needs to be collected and stored in a format that can later be used to train the supervised computer vision pain detection model.	User labels are stored in a relational Azure SQL database to be accessed easily from Python when doing model training and benchmarking.
Users want to track labeling process.	Reporting on top of the SQL relational data store of labeled data needs to be reported and visualized.	Power BI can be used to provide reporting on data labeling progress by displaying counts
Users want to see the IRR across users.	IRR is required at each frame level.	IRR calculations can use Python code on Azure Cloud Databricks to compute.
Funding allows a 30-day timeline.	Users require functionality, security, and authentication.	Cloud Native solution architecture can enable rapid delivery by leveraging pre-built components.

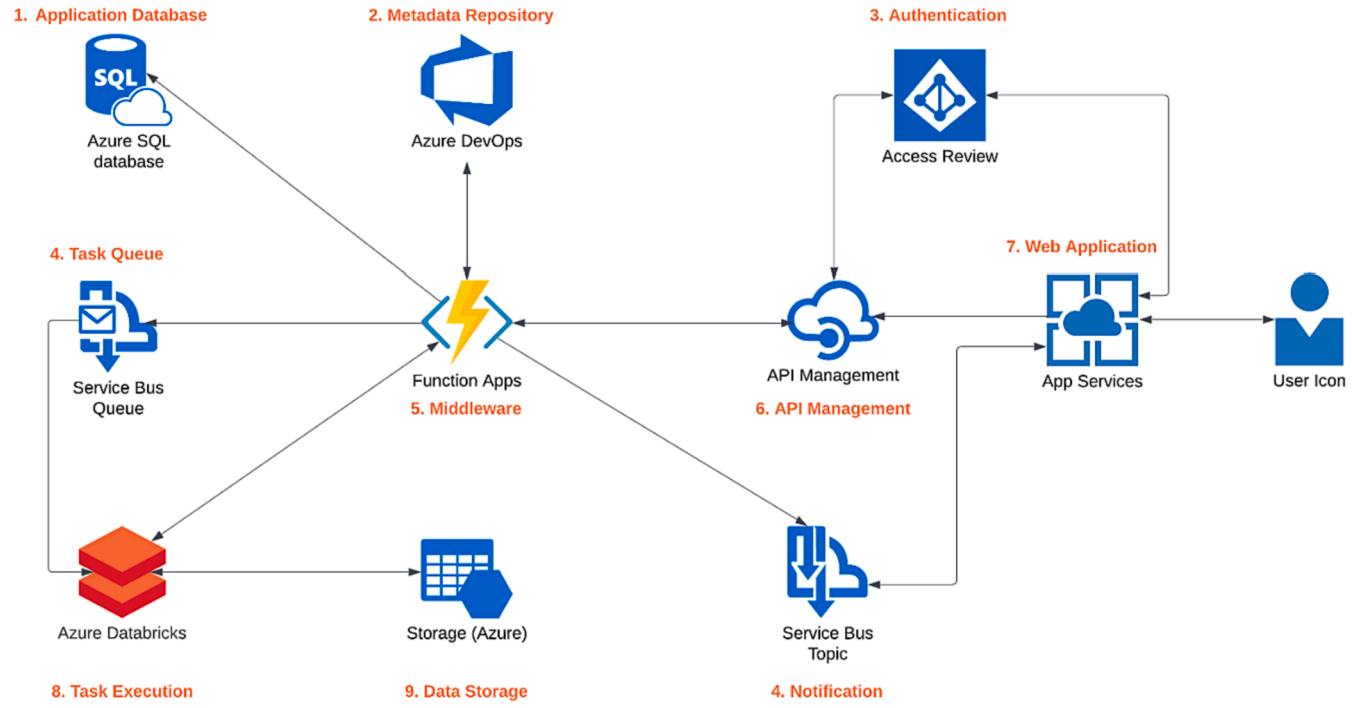
Landmark labeling took nurses a mean of 51.24 s per image and 20.36 min per video. In total, NFCS and landmark labeling took nurses a mean of 75 s per image and 29.83 min per video. The best performing ML model from nurses' labeling of this data in H2AI had 97.7 % precision, 98 % accuracy, 98.5 % recall, and Area Under the receiver operating characteristic Curve (AUC) of 0.98 [34]. HCDe-MLS and development of H2AI was a critical first step in the development of a trustworthy PRAMS.

5. Discussion

Our cross-functional team leveraged the HCDe-MLS model to develop the H2AI solution. H2AI is a data labeling solution that facilitates efficient labeling of video image data by SMEs and stores the user generated data labels for later development of and access by ML models. In this case, data labeled by nurses was used to train a highly precise and accurate model with excellent recall. With further refinement, H2AI will now be used to train an ML model to continuously monitor neonatal facial actions for pain, a Pain Recognition Automated Monitoring System (PRAMS).

5.1. ML models and efficacy comparison for pain classification

With 98 % accuracy, 97.7 % precision, 98.5 % recall, and AUC of 0.98, our supervised ML pain classification model far exceeded previously reported models developed with the same video dataset (highest



1. **SQL Database:** All applications writable data for operational purposes. Data is synced to storage.
2. **DevOps:** Git-based code repository storing all the metadata required by the application.
3. **Authentication & Authorization:** Admin & User role confirmation within the application, based on group designation in the directory.
4. **Service Bus:** Queue and Notification Services that control communication to Databricks as well as asynchronous notifications to the front end. Controls access to data storage, code repository, analytics services, operational database and endpoints.
5. **Function Apps:** Python serverless functions that can be scaled on demand. Middleware functions: (1) Front end services: support the front web application; (2) Repository Services: interact with the metadata; and (3) Analytics Services: interact with the Analytics backend.
6. **API Management:** Controls access to data storage, code repository, analytics services, operational database and endpoints using the authentication token.
7. **App Services:** Web App front end user interface built using Angular version 9. Communicates to components using Web Services.
8. **Databricks:** All analytics and data processing requirements of the application are performed by Azure Databricks. Analytic instances are created on demand, based on job submissions, and terminated once the job is completed. Jobs are invoked by middleware services, which are triggered by submissions to the job queue. Middleware services uses Databricks API to communicate. **Storage:** All application input and output data (csv, json, and other media files)

Fig. 4. H2AI solution architecture.

Fig. 5. Converting video images to frames for labeling tasks.

AUC = 0.93) and was better performing than all except one model developed with a smaller (AUC = 0.98, 15 videos) dataset [25,26,34]. As Zamzmi, et al. suggested [26] incorporating clinical and contextual information is necessary to refine and develop a context-sensitive PRAMS. Using HCDe-MLS and H2AI, we have demonstrated a method to effectively incorporate nurses' clinical and contextual knowledge to advance development of effective pain recognition models and PRAMS [34].

Using HCDe-MLS and H2AI also improved data labeling efficiency. Researchers using other methods in their attempt to automate pain assessment based on facial expressions have reported that data labeling was time and labor-intensive, taking up to 3 h for every minute of video [35]. Brahnam et al. [25] used iCOPEvid video images and Gaussian of Local Descriptors (GOLD) approach to extract facial features. This is a time-consuming four-step process that involves dense scale-invariant feature transform (SIFT) descriptors and probability density estimation. SIFT is computed based on the histogram of the gradient, making it mathematically complicated and computationally heavy.

Ashraf et al. [36] utilized the Active Appearance Model (AAM) to identify shape and appearance variations of adult faces but identified a lack of ground truth at the individual frame level. Also in contrast to our

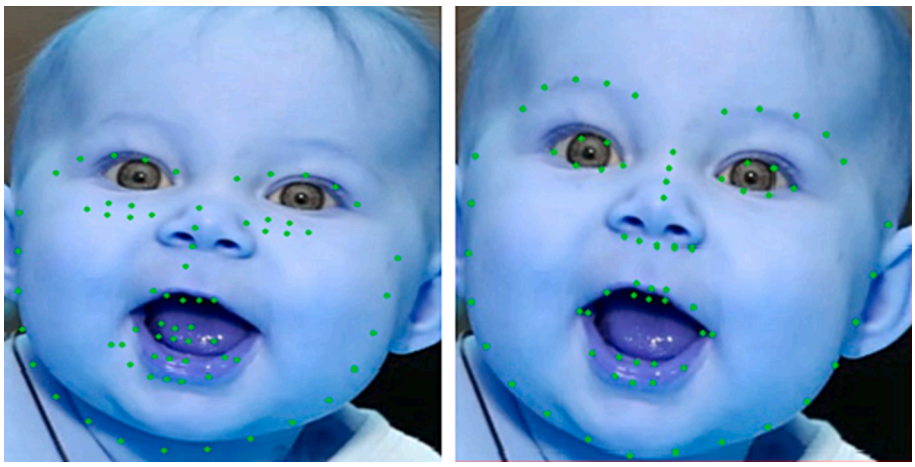


Fig. 6. Facial landmarking comparison by model: Caffe-based CNN (left) and MobileFaceNet (right).

approach, Brahnam et al. [25] achieved ground truth at the frame level and validated their neonatal pain classification ML model based on assessments by 185 college students with no appreciable healthcare or neonatal pain assessment experience. By having a frame-level ground truth based on data labeled by nurse SMEs, our model can learn and improve in its performance. This level of data labeling granularity is needed to ensure nurses will trust PRAMS, a CDSS solution for pain detection.

5.2. Limitations

H2AI and our best performing ML model was developed using the iCOPEvid neonatal pain database. This database is small and lacks racial and ethnic diversity [25] that may influence MobileFaceNet detection of facial landmarks [33]. Therefore, time required for data labeling by SME may be longer with a more diverse dataset. Recent federal data sharing requirements may facilitate access to more diverse video and clinical datasets that may accelerate further development of models that promote healthcare equity in CDSS and PRAMS.

The iCOPEvid database contained video that we then converted to frame images for data labeling granularity [25,26]. However, the resulting ML model may fail to capture dynamic patterns of facial expressions that may be important for discriminating pain or other conditions. To date, only one novel multimodal spatiotemporal approach for assessing neonatal postoperative pain has been reported with an AUC of 0.87 and 79 % accuracy, exceeding many other unimodal facial coding approaches [37].

5.3. 5.3 Future potential H2AI applications

H2AI can be utilized to label data and develop ML models to detect pain in other vulnerable patients who cannot provide self-report [24], to detect other human conditions associated with facial actions, such as depression and anxiety [38], and to detect potential threats by differentiating anger from hostility using micro-expressions [39]. With customization, H2AI can also be extended to Natural Language Processing (NLP) models, where the model is trained to deliver sentiment analysis, entity name recognition, and optical character recognition. Audio tagging is also a potential area of development for H2AI, such that information pertaining to the sound bites from the videos, such as cry, could assist in the model's learning process. We are moving forward to develop PRAMS with a clinical trial of continuous video facial monitoring for pain. Determining the latency of alert, specifically, the length of time or number of consecutive images that classify a condition before a clinician is alerted, is a feature we must add to H2AI.

6. Conclusions

When training computer vision algorithms for healthcare CDSS, ML models must be explainable and validated against the expertise of healthcare professionals. We have demonstrated that HCDe-MLS can be used to generate a user-centric software solution with embedded ML. We engaged nurses in the design, building, and deployment of H2AI, a first step in our development of a PRAMS. To meet nurses' needs and deliver the best user experience, we used Cloud Native, a serverless architecture to accelerate time to solution delivery. OpenCV provided efficient video-to-image data pre-processing for data labeling. MobileFaceNet demonstrated superior results for default landmark placement on neonatal video images. We found that H2AI facilitates efficient data labeling and stores labeled training data for future access to train ML models. H2AI also tracks IRR and compares ML model performance to SMEs. The H2AI solution can be generalized to other industry uses.

Summary Table:

What is already known on the topic:

- Individual factors are most important for influencing perceptions of software quality.
- User-interface design, perceived usefulness, helpfulness, functionality, reliability, and ease of use, as well as security/service-related and social factors influence trust and clinical decision support software adoption.

What this study added to our knowledge:

- The Human-Centered Design for Embedded Machine Learning Solutions (HCDe-MLS) model provides a systematic approach for engaging nurses to develop patient monitoring clinical decision support software solutions.
- Nurses informed the development of Human-to-Artificial Intelligence (H2AI), an intuitive and efficient data labeling software solution for healthcare professionals' use.

7. Financial disclosure statement

This study was supported by a Perinatal Origins of Disease Grant from Stanley Manne Children's Research Institute and The National Science Foundation grant number #2205472. Dr. Manworren is supported by The Posey and Fred Love Endowment of Nursing Research at Ann & Robert H. Lurie Children's Hospital of Chicago.

CRediT authorship contribution statement

Naomi A. Kaduwela: Conceptualization, Software, Validation, Formal analysis, Investigation, Resources, Data curation, Writing – review & editing, Visualization, Supervision, Project administration, Funding acquisition. **Susan Horner:** Validation, Formal analysis, Investigation, Resources, Writing – review & editing, Visualization, Supervision, Project administration. **Priyansh Dadar:** Software, Validation, Formal analysis, Data curation, Writing – review & editing. **Renee C.B. Manworren:** Validation, Formal analysis, Investigation, Resources, Writing – original draft, Writing – review & editing, Visualization, Supervision, Project administration, Funding acquisition.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

We would like to thank Keri Benbrook, Ashley Entler, Taylor Greene, Catherine Myler, Karah Riley, Kim Rindeau, and Rebecca Zuravel, bacclaureate degree prepared NICU registered nurses at Ann & Robert H. Lurie Children's Hospital of Chicago for their participation in focus groups to developed H2AI and labeling of data to train the ML model for PRAMs. We would also like to thank Rajesh Inbasekaran and Balakumaran Manoharan for their guidance in overall software solution architecture and application design and development, and Rahul Dhanasiri for his help in building the MobileFaceNet model. Lastly, we would like to acknowledge Jun Hua Wong, Kosha Soni, Trishla Mishra, and Stacey Tobin for their assistance in literature review, manuscript preparation, and editing.

References

[1] R.C.B. Manworren, Nurses' management of children's acute postoperative pain: a theory of bureaucratic caring deductive study, J. Ped. Nurs. 64 (2022) 42–55, https://doi.org/10.1016/j.pedn.2022.01.021.
[2] G.A. Boy, Human-centered design of complex systems: an experience-based approach, Design Sci. 3 (2017) e8, https://doi.org/10.1017/dsj.2017.8.
[3] M. Melles, A. Albayrak, R. Goossens, Innovating health care: key characteristics of human-centered design, Int. J. Quality Health Care 33 (1) (2020) 37–44, https://doi.org/10.1093/intqhc/mzaa127.
[4] T.M. Ward, M. Skubic, M. Rantz, A. Vorderstrasse, Human-centered approaches that integrate sensor technology across the lifespan: opportunities and challenges, Nurs. Outlook 2020 (2020) 734–744, https://doi.org/10.1016/j.outlook.2020.05.004.
[5] D. Zikos, A framework to design successful clinical decision support systems. Proceedings of the *10th International Conference on Pervasive Technology Related to Assistive Environments.* In: PETRA 2017. Association for Computing Machinery, New York, NY, USA, 2017, 185-188. DOI: 10.1145/3056540.3064960.
[6] Canadian Standards Association, CAN/CSA-ISO/IED 25050:12, Systems and software engineering- Systems and software quality requirements and evaluation (SQuaRE-System and software quality models), 2012. http://scc.ca/en/standadsdb/standards/28356.
[7] ISO 25000 STANDARDS, 2019. https://iso25000.com/index.php/en/iso-25000-standards.
[8] L. Souza-Pereira, S. Ouhbi, N. Pombo, Quality-in-use characteristics for clinical decision support system assessment, Comput. Methods Programs Biomed. 207 (2021) 106169, https://doi.org/10.1016/j.cmpb.2021.106169.
[9] K. Curcio, A. Malucelli, S. Reinehr, M.A. Paludo, An analysis of the factors determining software product quality: a comparative study, Comput. Standards Interfaces 48 (2016) 10–18, https://doi.org/10.1016/j.csi.2016.04.002.
[10] M.B. Garcia, N.U. Pilueta, M.F. Jardiniano, VITAL APP: Development and user acceptability of an IoT-based patient monitoring device for synchronous measurements of vital signs, in: IEEE 11th International Conference on Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment, and Management (HNICEM), Laoag, Philippines, 2019, pp. 1–6. DOI: 10.1109/HNICEM48295.2019.9072724.
[11] S. Pengate, P. Antonenko, A multimethod evaluation of online trust and its interaction with metacognitive awareness: an emotional design perspective, Int. J. Human-Comput. Interaction 29 (2013) 582–593, https://doi.org/10.1080/10447318.2012.735185.
[12] A. Anand, A.S. Mishra, A. Deep, K. Alse, Generation of educational technology research problems using design thinking framework, in: 2015 IEEE Seventh International Conference on Technology for Education (T4E), 2015, pp. 69–72, https://doi.org/10.1109/T4E.2015.28.
[13] H. Suresh, J. Guttag, A framework for understanding sources of harm throughout the machine learning life cycle (2021). arXiv:1901.10002V4. DOI: 10.48550/arXiv.1901.10002.
[14] E.V. Eikey, M.C. Reddy, C.E. Kuziemsky, Examining the role of collaboration in studies of health information technologies in biomedical informatics: a systematic review of 25 years of research, J. Biomed. Inform. 57 (2015) 263–277, https://doi.org/10.1016/j.jbi.2015.08.006.
[15] K. Thoring, R. Muller, Understanding design thinking: a process model based on method engineering, in: 13th International Conference on Engineering and Product Design Education, London, UK, 8-9 September 2011, 2011, pp. 493-498. The Design Society. https://www.designsociety.org/publication/30932/.
[16] A.A. Abahussin, R.M. West, M.J. Allsop, D.C. Wong, L.E. Ziegler, A pain recording system based on mobile health technology for cancer patients in a home setting: a user-centred design, in: 2020 IEEE International Conference on Healthcare Informatics (ICHI), 2020 (11) 1–10. DOI: 10.1109/ICHI48887.2020.9374388.
[17] G. Tobias, A.B. Spanier, Developing a mobile app (iGAM) to promote gingival health by professional monitoring of dental selfies: user-centered design approach, J. Med. Internet Res. 8 (8) (2020) 17, https://doi.org/10.2196/19433.
[18] R. Ashmore, R. Calinescu, C. Paterson, Assuring the machine learning lifecycle: desiderata, methods, and challenges, ACM Comput. Surv. 54 (5) (2021) 111, https://doi.org/10.1145/3453444.
[19] C.F. Hsu, C.C. Lin, T.Y. Hung, C.L. Lei, K.T. Chen, A detailed look at CNN-based approaches in facial landmark detection (2020). arXiv:2005.08649. DOI: 10.48550/arXiv.2005.08649.
[20] F. Liang, W.G. Hatcher, W. Liao, W. Gao, W. Yu, Machine learning for security and the internet of things: the good, the bad, and the ugly, IEEE (2019), https://doi.org/10.1109/ACCESS.2019.2948912.
[21] H. Kuwajima, H. Yasuoka, T. Nakae, Engineering problems in machine learning systems, Mach. Learn. 109 (2020) 1103–1126, https://doi.org/10.1007/s10994-020-05872-w.
[22] E.P.S. Baumer, Toward human-centered algorithm design, Big Data Soc. 4 (2) (2017) 1–12, https://doi.org/10.1177/2053951717718854.
[23] B. Strauch, The automation-by-expertise by-training-interaction: why automation-related accidents continue to occur in sociotechnical systems, Hum. Factors 59 (2) (2017) 204–228, https://doi.org/10.1177/001872081666545.
[24] K. Herr, P. Coyne, B. Ely, C. Gelinas, R.C.B. Manworren, Pain assessment in the patient unable to self- report: clinical practice recommendations in support of the ASPMN 2019 position statement, Pain Manag. Nurs. 20 (5) (2019) 404–417, https://doi.org/10.1016/j.pmn.2019.07.005.
[25] S. Brahnam, L. Nanni, S. McMurtrey, A. Lumini, R. Brattin, M.R. Slack, T. Barrier, Neonatal pain detection in videos using the iCOPEvid dataset and an ensemble of descriptors extracted from gaussian of local descriptors, Appl. Comput. Inf. 19 (2020) 122–143, https://doi.org/10.1016/j.aci.2019.05.003.
[26] G. Zamzmi, R. Kasturi, D. Goldgof, R. Zhi, T. Ashmeade, Y. Sun, A review of automated pain assessment in infants: features, classification tasks, and databases, IEEE Rev. Biomed. Eng. 11 (2018) 77–94, https://doi.org/10.1109/RBME.2017.2777907.
[27] R.V.E. Grunau, K.D. Craig, Pain expression in neonates: facial action and cry, Pain 28 (1987) 395–410, https://doi.org/10.1016/0304-3959(87)90073-X.
[28] L.M. Relland, A. Gehred, N.L. Maitre, Behavioral and physiological signs for pain assessment in preterm and term neonates during nociception specific response: a systematic review, Pediatr. Neurol. 90 (2019) 13–23, https://doi.org/10.1016/j.pediatrneurol.2018.10.001.
[29] S. Emami, V.P. Suciu, Facial recognition using OpenCV, J. Mobile Embedded Distrib. Syst. 4 (2012). https://www.researchgate.net/publication/267426877_Facial_Recognition_using_OpenCV.
[30] J. Deng, J. Guo, N. Xue, S. Zafeiriou, Arcface: additive angular margin loss for deep face recognition, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 2019, pp. 4685-4694. DOI: 10.1109/CVPR.2019.00482.
[31] S. Chen, Y. Liu, X. Gao, Z. Han, Mobilefacenets: efficient CNNs for accurate real-time face verification on mobile devices, in: Chinese Conference on Biometric Recognition, Springer, Cham, 2018, pp. 428–438. DOI: 10.1007/978-3-319-97909-0_46.
[32] G.B. Huang, M. Mattar, T. Berg, E. Learned-Miller, Labeled faces in the wild: a database for studying face recognition in unconstrained environments, in: Technical Report, 07-49. University of Massachusetts, Amherst, 2007. http://vis-www.cs.umass.edu/lfw/lfw.pdf.
[33] H. Wang, H. Zhang, L. Yu, L. Wang, X. Yang, Facial feature embedded cyclegan for Vis-Nir translation, in: ICASSP 2020 – 2020 IEEE International Conference on Acoustics, Speech and Signal Processing. (ICASSP), 2020, 1903–1907. DOI: 10.1109/ICASSP40776.2020.9054007.
[34] R.C.B. Manworren, S. Horner, R. Joseph, P. Dadar, N. Kaduwela, Performance evaluation of a supervised machine learning pain classification model developed by neonatal nurses, Adv. Neonatal Care (*accepted for publication,* 2023).
[35] J.W.B. Peters, H.M. Koot, R.E. Grunau, et al., Neonatal facial coding system for assessing postoperative pain in infants: item reduction is valid and feasible, Clin. J. Pain 19 (2003) 353–363.
[36] A.B. Ashraf, S. Lucey, J.F. Cohn, T. Chen, Z. Ambadar, K.M. Prkachin, P. E. Solomon, The painful face–pain expression recognition using active appearance models, Image Vis. Comput. 27 (12) (2009) 1788–1796, https://doi.org/10.1016/j.imavis.2009.05.007.

[37] M.S. Salekin, G. Zamzmi, D. Goldgof, R. Kasturi, T. Ho, Y. Sun, Multimodal spatio-temporal deep learning approach for neonatal postoperative pain assessment, Comput. Biol. Med. 129 (2021) 104150.
[38] M. Gavrilescu, N. Vizireanu, Predicting depression, anxiety, and stress levels from videos using the Facial Action Coding System, Sensors (Basel) 19 (17) (2019) 3693, https://doi.org/10.3390/s19173693.
[39] M. Tsikandilakis, P. Bali, J. Derrfuss, P. Chapman, Anger and hostility: are they different? An analytical exploration of facial-expressive differences, and physiological and facial-emotional responses, Cogn. Emotion 34 (3) (2019) 581–595, https://doi.org/10.1080/02699931.2019.1664415.