RecurrentCare: Integrating XAI into Automated Diagnostics of Knee Osteoarthritis

Recurrent Care

Antonios Matzakos Chorianopoulos (13915320), Carl-Phillip Senze (11394269), Carmen Byrne Salsas (12968900), Bartosz Smoczyński (13898183), Max Winder (13881590) and Casper Dahmen (11887370)
Data System Projects 2021-2022
University of Amsterdam

ABSTRACT

Knee osteoarthritis (OA) is one of the most frequent musculoskeletal disorders. While there is no cure, early treatment can improve the quality of life of OA patients. Currently, knee OA severity is determined after the assessment of patient's X-rays. We propose a method to assess knee osteoarthritis severity grading according to the Kellgren-Lawrence scale by means of Convolutional neural networks (CNN). We trained four different architectures achieving 71% multi-class accuracy with DenseNet201. Our aim was to provide an AI tool with a transparent physician-oriented design. Thus, we implemented four XAI methods; namely, Integrated Gradients, Guided Backpropagation, Guided Backpropagation with local variance and LIME. We used Streamlit in order to make this tool available to physicians in a simple web application. This application was validated through a Think Aloud protocol and the SUS questionnaire. The validation process revealed four severe problems. The SUS score reached 79.5 which is considered good, with a low level of certainty.

KEYWORDS

Knee Osteoarthritis, Convolutional Neural Networks, Explainable AI, Integrated Gradients, Guided Backpropagation with Local variance, LIME, automated diagnosis

1 INTRODUCTION

For elderly people, living with a disability is a major issue. According to [1], nearly half of all aged people (those over the age of 60) have a serious impairment with arthritis being one of the most common causes. Just in the United States nearly 50% of all adults suffer from a form of arthritis [2]. According to the Osteoarthritis Action Alliance over 242 million people suffer from hip or knee osteoarthritis. This leads to an estimated cost of $136 billion annually [3]. Osteoarthritis and arthritis belong to the same pathological group of rheumatism but have different causes and symptoms.

Knee Osteoarthritis

Osteoarthritis describes a joint sickness that is caused by diminishing cartilage in the knee joint. This process is caused by classic "wear-and-tear" processes over the years [4]. In contrast to that, normal arthritis is caused by an inflammation which gradually destroys the structure of the joints. The diagnosis of knee osteoarthritis commonly is done with a initial physical examination of the knee and then confirmed with a radiological method [4]. While methods such as computed tomography (CT), ultrasound and magnetic resonance imaging (MRI) are commonly used, these methods are not suitable for the actual diagnosis but rather for analysis of soft tissues and fluid space or to rule out other diseases [5]. The main method to diagnose knee osteoarthritis is a radiograph (also referred to as X-ray). With the help of radiographs practitioners try to find typical characteristics of knee osteoarthritis. These are the formation of new bone along the joints (osteophytes), the narrowing of joint space and other abnormalities in the lower end of the two joint bones such as sclerosis, cysts, shape changes and loss of bone volume [5].

Kellgren and Lawrence System

Knee osteoarthritis can have several stages of disease progression and is not curable. This progression is often quantified with the Kellgren and Lawrence System [6] (KL). With the help of this system it is possible for practitioners to quantify the severity of the knee osteoarthritis.

Grade	Severity
0	None
1	Doubtful
2	Minimal
3	Moderate
4	Severe

Table 1: Kellgren-Lawrence Scale [4]

As we can see in Table 1, there are five severity stages. In grade 0 there are no visible causes for osteoarthritis. For grade 1 a clinician identifies some doubtful joint space narrowing (JSN) and possible osteophytic lipping. With grade 2 the diagnosis for the two previous causes can be confirmed, leading to a definite diagnosis of osteoarthritis. A patient has grade 3 osteoarthritis when there are moderate multiple osteophytes, definite JSN, some sclerosis and possible bone end deformity. The last and most severe grade 4 is given, if the clinician can identify large osteophytes, marked JSN, severe sclerosis and deformity of bone ends [4]. Even though this description provides indicators to come to a final diagnosis there are some practical problems associated with this scale. Studies have shown that regardless of their experience doctors do not seem to classify knees correctly according to the KL-System. These studies show that clinicians do not seem to agree with each other, resulting in a low inter-rater reliability based on Cohen's kappa metric [7]. Next to that, clinicians struggle to find certain causes of osteoarthritis, such as osteophytes, in a picture [8]. Taking into account the enormous costs that osteoarthritis causes, especially with regards to diagnosis and treatment, and problems in relation to the Kellgren and Lawrence system we propose an automated system based on deep neural networks to support clinicians in their decision making.

Explainable Computer Aided Diagnostics

Current state-of-the art methods use convolutional neural nets to predict the severity of knee osteoarthritis on the KL-Scale [9]. While these systems do perform very well, they are black-box systems that do not show how the decision was made. This can impact the doctor-patient relationship heavily as the diagnosis is transferred from the certified doctor to an entity that the patient cannot assess anymore [10]. Therefore, the objective of this report is to implement explanation methods that show how the decision of the system was made in an understandable way. Here it is very important to have a method that is consistent. We propose a system that consists of an automated diagnosis, an explanation mechanism and a visualisation of this explanation through an application with a simple, physician-oriented design.

2 RELATED WORK

Automated Diagnostics

Traditionally, knee osteoarthritis radiographs are analysed by musculoskeletal radiologist. Early approaches to automated diagnostics used explicit feature extraction to gain information about the given X-ray and then predicted the severity with traditional classification methods [11]. More recent methods rely on convolutional neural networks and can achieve human-like performance [9, 12, 13]. As knee osteoarthritis is not curable the repeated diagnosis is costly in terms of time and money. Therefore, automated tools can mitigate human bias and costs in osteoarthritis treatment [13, 14].

Explainable AI

In general neural networks are regarded as black-box models. This means that it is not possible to instantly understand how the system made a decision. Instead, the decision comes out of a "black-box". Due to the unexplainability of neural networks a demand arose for methods which explain whether the result was based on, in our case clinically, relevant information. *Explainable* AI (XAI) methods can be applied to make such systems more interpretable and understandable [15]. Van der Velden et al. [16] show that saliency maps can be generated to reveal parts of an image that were important in determination of the specific score. Saliency maps are also called visual explanations and are the major method used in this report. There are three types of approaches to visual explanations: backpropagation-based, perturbation-based or multiple instance learning-based approach following Van der Velden et al 2021 [16]. For this application the focus was on backpropagation- and perturbation-based methods.

Human in the loop

Even though convolutional neural networks can achieve impressive results, there are some problems in the medical field which are difficult to be solved using solely machine learning. The quality

of the results might be questionable or to make the application fully automated would be hard to do. An example where humans outerperform machine learning algorithms are in radiologic imaging. The cases where the experts can't be detained from such applications using machine learning do need an integration of the experts with the data. The experts can interact with the machine learning application which is called interactive machine learning (iML). IML puts the "human-in-the-loop" (HITL) to enable the collaboration of humans and computers to achieve what neither a human nor a computer could do on their own. These iML approaches ensure the optimization of the learning behaviour through the interactive setup [17].

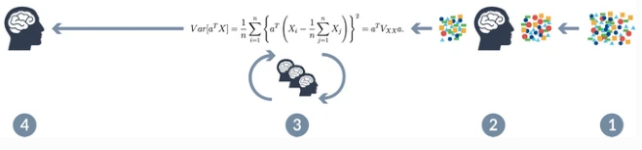

Figure 1: HITL machine learning pipeline [17]

In the above figure the machine learning pipeline using HITL is showed. 1: the input data. 2: the pre-processing phase. 3: the expert interacts with machine learning, 4: output is checked by the expert.

Think Aloud

Jakob Nielsen, who became known as the "guru of Web page usability" in the late 90', considers the Think Aloud method to be the "single most valuable usability engineering method". Three steps are required to implement this usability test [18] :

- Recruit representative users.
- Give the users individual tasks to perform.
- Remain quiet and prompt the users to share their thoughts aloud while they explore the application.

According to the literature only 5 experts are required in the Think Aloud sessions to discover over 80% of the usability problems of an application. This method may suffer from bias, since the situation is unnatural for the interviewees. Nonetheless, it is difficult to obtain unreliable results with this method and it is accessible to most since no special equipment is required [19].

3 METHODS

Dataset

The dataset we used for this project provides X-ray images of knees [20]. It is an excerpt from the Osteoarthritis Initiative, which is a longitudinal study to find new ways to detect biomarkers for osteoarthritis as indicators for disease beginning and progression. Over 4,000 participants between the ages of 45-79 participate in this study and 58% are female. Additionally, all ethnicities are represented in the study with a focus on African-Americans [21].

Data Preprocessing & Augmentation

Many Computer Vision tasks have shown that deep CNNs are exceptionally accurate. However, in order to prevent overfitting, these networks rely on large datasets. Overfitting occurs when a network develops a function with extremely high variance in order to perfectly model the training data. In many application fields, such as medical image processing, practitioners do not have access to large datasets. Data Augmentation is a process that alludes to a range of strategies to the improve on the size and quality of training datasets so that better Deep Learning models may be generated [22].

Random Erasing. For training a CNN, a new data augmentation method has been developed. Random Erasing picks a rectangular region in a picture at random and erases its pixels with random values during training. This procedure generates training images with varying degrees of occlusion, reducing the danger of overfitting and making the model occlusion-resistant. Random Erasing does not need parameter learning, is simple to build, and can be used with nearly any CNN-based recognition model. Random Erasing, despite its simplicity, is a useful supplement to standard data augmentation techniques like random cropping and flipping, and it consistently improves image categorization, object detection, and person re-identification when compared to strong baselines [23].

Horizontal & Vertical Flip. In the case of a vertical or horizontal flip, an image flip entails reversing the rows or columns of pixels. A boolean horizontal flip or vertical flip option to the ImageDataGenerator class constructor specifies the flip augmentation.

Shear Range. The image's shape is angled when a shear transformation is used. Shear transformation differs from rotation in that it fixes one axis while stretching the image at a specific angle known as the shear angle. This results in a stretch in the image that is not visible in rotation. The angle of the slant in degrees is specified by the shear_range [24].

Rotation Range. The data generated is rotated randomly by a given angle when the rotation range parameter is used.

Width Shift Range. The width shift range is a floating-point value between 0.0 and 1.0 that determines the upper bound of the fraction of the overall width by which the picture will be randomly shifted left or right.

Height Shift Range. Instead of being horizontally displaced, the image is vertically shifted.

Zoom Range. The zoom range argument is used to get a random zoom before feeding the image in the network.

CNN Architectures

All models used by us fall into the category of convolutional neural networks (CNNs). A CNN gets its name from the use of convolutional layers. In simple words such networks use image filters to detect relevant features in an image. In this paper we used the following architectures:

DenseNet201. This model uses *dense blocks* i. e., blocks of pairwise connected layers. This reduces the number of parameters, combats the vanishing gradient problem and encourage feature reuse [25].

ResNet152V2. Resnets introduce *residual connections* i. e., inter-layer connections which redefines the learning process as that of learning residual functions. It causes the gradient to flow deeper into the network allowing for much more layers [26].

InceptionV3. This model uses all the traditional components of a CNN, but its architecture is carefully optimized. The local assembling of layers is generalized spatially to the whole network. [27].

Xception. The architecture of this model resembles that of an inception network with the difference, that inception modules have been replaced with depth-wise separable convolutions [28].

Transfer Learning & Fine- tuning

Transfer learning and fine tuning are defined as the process of training a model on new data while initializing it with pretrained weights obtained from training it on a previous dataset. For all the architectures in this study we used pretrained weights of the models based on the imagenet-dataset [29] to initialize our models. Additionally, we fine-tuned all models by adding global average pooling, dropout and dense layers on top of the base model. The dropout layer makes the training process noisy by requiring nodes within a layer to take on, more or less responsibility for the inputs on a probabilistic basis. Dropout, breaks up situations where network levels co-adapt to remedy faults made by previous layers, making the model more robust [30]. As the KL-scale has five categories we the last layer of the network is a dense layer with five output neurons and a softmax-function get class-probabilities. We hoped that by using techniques that took the complexity of our weights into consideration during optimization, we may direct the networks toward a more general, but scalable, mapping rather than a very data-specific one. Hence, we used L1-Lasso L2-Ridge, known as elastic net regularization.

Training Set-Up & Hyperparameter tuning

Tuning the hyperparameters is a crucial step in the process of designing a well performing model. These are parameters that should manually be estimated. This process cannot be methodized and it is highly contingent on the data we use.

Loss Function. We used categorical cross-entropy loss, because we had to deal with a multi-class classification problem.

Activation Function. We employed the Softmax Activation function in the output layer of the CNN models to predict with a multinomial probability distribution.

Optimizer & Learning Rate. The optimizer option specifies the algorithm that will be used to optimize our model. We used Adam optimizer with a learning rate of 0.00001.

Batch Size. In this study we selected a batch size of 32 images. The batch size is the number of training examples utilized in one forward pass.

Training Epochs. We chose to train our models for 103 training epochs(forward and backward propagation) and saved the best weights during the training procedure.

Guided Backpropagation

XAI methods in image analysis focus on visualizing which parts of the input constituted most to the classification decision of a model. The probability for a class in a neural network is given by the activation a of a single neuron corresponding to that class. This activation is a differentiable function of the pixel intensities in the input $a = f(\vec{x})$, where $\vec{x}$ is the input. Therefore, it is natural to consider the gradient of the neuron with respect to the input $\frac{\partial f}{\partial x}$. Recall, that from the definition of a derivative it follows, that the small changes in the arguments with high gradients will have the greatest impact on the value of a function. Thus calculating $\frac{\partial f}{\partial x}$ should highlight pixels that were crucial to the classification decision of the neural network. However as can be seen in the figure the resulting image isn't satisfactory. An improvement of the technique was proposed in [31]. The difference is that when backpropagating through the ReLU activation functions only the positive gradients are backpropagated. The result is that only the features of the input that are positively correlated with the predicted class are highlighted.

Local Variance

Outputs of the guided backpropagation have high resolution but lack readability. This is why we propose a post-processing procedure which we call **local variance**. For an image $\vec{x}$ let $window(\vec{x}, p, l)$ denote a square sub-array of $\vec{x}$ centered at p with side length l measured in pixels, where p is a point in the image. Then we define $localVariance(\vec{x})_p = Var(window(\vec{x}, p, l))$, where l is a parameter (for our case we chose $l = 20$). As can be seen in the figure this results in highlighting the regions of the image where the most "action" happens with regard to the guided backpropagation output. Additionally, we noticed that guided backpropagation tends to produce visual artifacts near the borders of the image. That's why for the final output or our method we manually zeroed-out the areas outside a predefined centered circle.

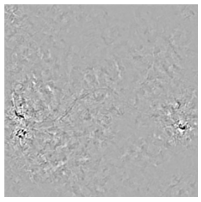

Figure 2: Vanilla Backpropagation

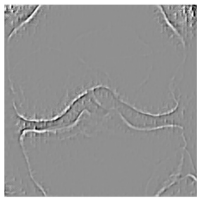

Figure 3: Guided Backpropagation

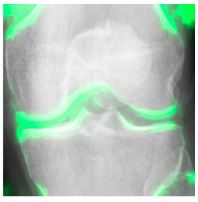

Figure 4: Local Variance

Integrated Gradients

Integrated Gradients (IG) is a deep neural network explainability approach that visualizes the relevance of input features that contribute to the model's prediction, a method for attributing the prediction of a classification model to its input characteristics. It works by calculating the gradient of the prediction output in relation to input characteristics. To find features a neural network considers essential with integrated gradients, it is necessary to define a baseline input. For computer vision applications usually dark image (all pixel values set to zero) or random noise can be used as a baseline input. For a certain number of steps, defined with α, we interpolate between the baseline (x') and the original image and (x).

$$x = x' + \alpha(x - x')$$

Then we calculate the gradients to understand how changes to a feature influence the model's prediction. In a final step we approximate the importance of a feature, in this case a pixel, by averaging gradients [32].

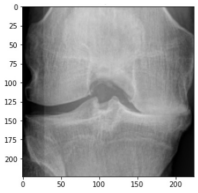

Figure 5: KL4 Ground Truth

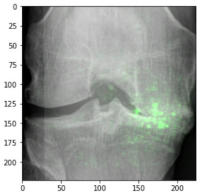

Figure 6: Normal Gradients

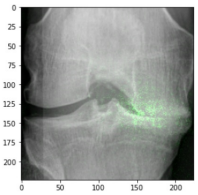

Figure 7: Integrated Gradients

LIME

LIME is model-independent, which means it may be used with any machine learning model. The technique tries to figure out what the model is doing by perturbating input images and seeing how

the predictions change. LIME generates explanations that indicate the contribution of each characteristic to the data sample prediction. Generally speaking, three steps are necessary to get visual explanations with LIME. First, it is necessary to generate a dataset of perturbed images. For image datasets this means random areas of an image are "greyed-out". Second, we obtain predictions for those images with the previously trained neural network. Third, a locally linear model is learned to identify mistakes made in perturbed images. As a last step the visual explanation is generated by presenting the most important pixels with the highest weights and omitting other parts of the image [33]. The final result can be inspected in Figure 8 as the original input image and Figure 9 as the XAI-output of the LIME method.

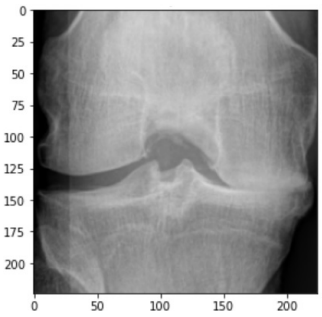

Figure 8: KL4 Ground Truth

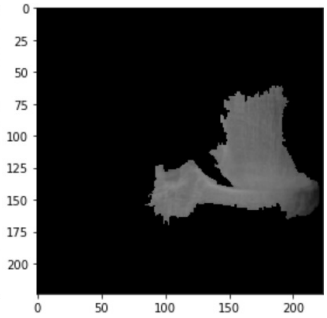

Figure 9: LIME Explanation

Front-end

Formative research. During the starting phase of the prototype, we invited a general practitioner with previous experience in several medical applications. The physician was presented with an explanation of the use case of the application and they were asked to navigate the interface freely. During this session the physician provided us with feedback on the application.

Pre-validation interface. The development of the interface for the Recurrent Care application was implemented using the Python library Streamlit [34]. Streamlit is growing in popularity amongst machine learning experts. This library helps developers link their models to interfaces with plug-and-play feautures. For this project, we used Streamlit's MultiApp feature which includes a default navigation bar to browse through the application's pages. We developed three pages guiding the physician through the information flow of our back-end models and their outputs: **Home > Upload your Data > Prediction**. The second page, "Upload your Data" is pointed at a specific directory; e.g., the Electronic Patient Record. By entering the patient number, a scraping function searches through the files named after that patient number in that directory and uploads them to the interface. Additionally, we used `streamlit.file_uploader()` as a tool to enable physicians to locally browse and upload the files stored in their personal folders. X-rays are typically stored as binary images, since they are often black and white. Since the back-end models expect images in the RGB space as input, the uploaded images where converted into RGB images using the `Image.convert()` tool from the Python library Pillow. The images are locally stored in the application's directory. These images are then opened again in the last page "Prediction" . In this page, the front-end is linked to the back-end. Checkbox menus for prediction models and XAI methods where created using `streamlit.radio()`. The input is then passed as an argument to a function running the chosen model and XAI method. This function returns the predicted KL score, and the confidence of the prediction per KL class. In addition, if the physician does not select "None" as an XAI method, the function returns the output image from the chosen XAI method. On the same page, we used `streamlit_drawable_canvas` to superimpose X-ray images with an interactive canvas where circles, rectangles, and free shapes can be drawn by the doctor. Additionally, a slide bar was created using `streamlit.slider()` for the severity grading, and a text input box was created using `streamlit.text_input()` to write the diagnosis. Plots were created using `streamlit.bar_chart()`, and a download button was created using `streamlit.download_button()`. Finally, since Streamlit reruns the entire script whenever new input is added from the interface, we saved the data from each page using `streamlit.session_state()` so that navigation from page to page would be possible without experiencing any data loss.

Human-in-the-loop. In this application experts can interact with the machine learning application, which is called human-in-the-loop (HITL) see figure 1 for more explanation. The expert can run the model to get the predicted severity grading for the X-ray. For all the possible grades the expert can observe how certain the model is per class

in percentages. If the model outcome has a high entropy the expert can use its own expertise to grade the severity of the knee by using the slider mentioned above. Or the expert gives the severity grading for every new incoming X-ray using the slider. The advantage is that there will be more new labeled X-rays to train the models, which will optimize the networks performance [35].

User Validation

An important aspect of making an application is to measure its usability. Measuring and tracking the usability is part of our summative research to validate, and eventually improve the user experience (UX) design of our app. The paper 'An empirical evaluation of the system usability scale (SUS)' written by A. Bangor et al. [36] is used as a grounding point for our usability evaluation. We evaluated the UX design of the Recurrent Care App in the two steps described below.

Think Aloud Protocol. We recruited five experts in Medical Informatics to participate in five individual Think Aloud sessions. The sessions were conducted over the video call platform Zoom, and the experts were given remote control of the screen in order to explore the Recurrent Care application. During the session, we explained the purpose of the Think Aloud session. We asked the experts to sign the informed consent forms acknowledging that the sessions would be recorded for further analysis and deleted at the end of the study. Next, we asked them to perform a mock task (i.e., opening the browser and searching for a calculator, then using the calculator). Finally, the experts were asked to perform the following 11 tasks:

(1) Read the Home page and input an image with patient number "000".
(2) Switch to the prediction tab, and draw circles on the uploaded images. Input a severity score. Input a diagnostic text.
(3) Go back and forth one page.
(4) Select a model and an XAI method.
(5) Run the model
(6) Open the severity grading graph.
(7) Return to the previous page and upload image by browsing the local files.
(8) Switch to the prediction tab, and draw rectangles on the uploaded images. Input a severity score. Input a diagnostic text.
(9) Select a different model and a different XAI method.
(10) Run the model.
(11) Download the XAI image.

Severity Grade	Interpretation
0	Not a usability problem
1	Cosmetic usability problem: need not be fixed unless extra time is available on the project
2	Minor usability problem: fixing this should be given low priority
3	Major usability problem: important to fix, should be given high priority
4	Usability catastrophe: imperative to fix this before product can be released

Table 2: Usability Problem Severity grading[37]

The experts were encouraged to share their thoughts aloud throughout the process. Following the completion of these tasks, our team revisited the recordings of the videos and transcribed the comments of the experts that were related to usability errors. In addition, we paid attention to the movement of the cursor on the screen in order to catch the issues that the experts may have not shared aloud. These were then coded into categories. We then calculated the number of tasks performed successfully. Finally, we gave each of the usability errors a severity score according to the following scale as shown in Table 2. The usability problems rated with a 4 were consequently corrected.

SUS score. At the end of the Think Aloud sessions the experts were invited to fill in the SUS questionnaire [36]. Since the experts filling in the questionnaire were medical informaticians and not physicians, we modified questions 1 and 7 to include the expectations of the application's usability from the perspective of a physician. The modified SUS questionnaire can be found on Table 3.

The point scale for the answers was as follows: **Strongly Disagree** (1 point), **Disagree** (2 points), **Neutral** (3 points), **Agree** (4 points), **Strongly Agree** (5 points). Finally, we calculated the SUS score with the following three steps [38]:

Number	Question
1	I think this app is well designed for doctors use.
2	I found the system unnecessarily complex.
3	I thought the system was easy to use.
4	I think that I would need the support of a technical person to be able to use this system.
5	I found the various functions in this system were well integrated.
6	I thought there was too much inconsistency in this system.
7	I would imagine that most doctors would learn to use this system very quickly.
8	I found the system very cumbersome to use.
9	I felt very confident using the system.
10	I needed to learn a lot of things before I could get going with this system.

Table 3: Modified SUS questionnaire

(1) Add the score for all odd-numbered questions. Subtract 5. Let X be the result of this step.
(2) Add the score for all even-numbered questions. Subtract this score from 25. Let Y be the result of this step.
(3) Let S be the SUS score: $S = (X + Y) \times 2.5$

Post-validation interface

Following the Think Aloud sessions and taking into account the insight gained from the results of the SUS questionnaire, the team fixed the most urgent usability issues.

4 RESULTS

Model Performance & Evaluation

As shown on Table 4, the accuracy of the four models used for the prediction of the KL score was greater than 0.68. In Figure 11 we show the confusion matrix of DenseNet201 per KL-score. The vertical axis represents the true values and the horizontal axis represents the predicted values.

Figure 12 shows several performance metrics per KL-score class for the same model.

Precision. The model's ability to avoid labeling a negative occurrence as positive.

Recall. The model's ability to correctly find all positive instances.

F1-Score. A weighted harmonic mean of precision and recall.

For all metrics 1.0 is the highest and 0.0 is the lowest. Weighted average is the simple mean of scores of all classes and macro average computes the metric separately for each class and then averages the results, therefore treating all classes equally. The main performance issues were observed in class 1 (KL 1).

CNN Architecture	Accuracy
DenseNet201	0.71
ResNet152V2	0.69
InceptionV3	0.68
Xception	0.68

Table 4: CNN Performance

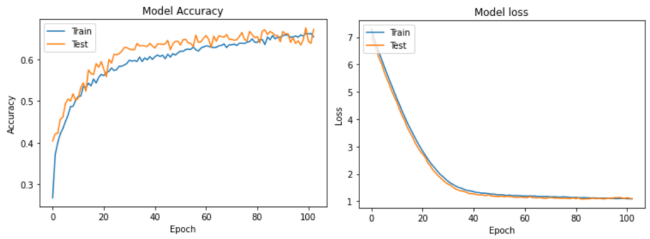

Figure 10: Training Progress - DenseNet201

Figure 10 shows the evolution of the accuracy and loss with respect to the number of epochs for the Xception architecture in the training and test sets. An increasing trend can be observed for the accuracy of the model in the training and test sets. Similarly, the training and test loss continue to decrease until 95th training epoch. Beyond this point, the model overfits the data: the model predicts extremely well on the training data, but it does not perform well on unseen data.

In each epoch we monitored the model's progress and saved the best weights based on the optimal validation accuracy.

XAI Methods

Guided Backpropagation. The outputs of guided backpropagation mostly emphasize the edges in the

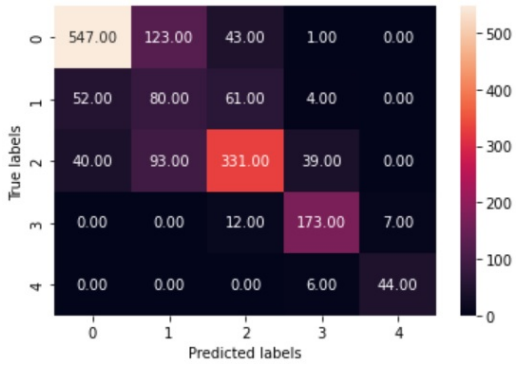

Figure 11: Confusion Matrix - DenseNet201

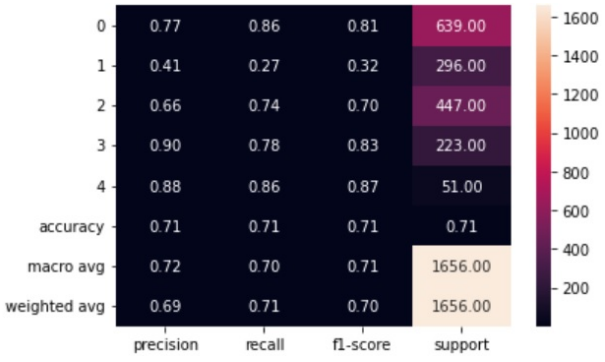

Figure 12: Classification Report - DenseNet201

images, especially in regions contributing to the final prediction. However some visual artifacts can also be observed, there is some noise visible in the corners of the picture. The method correctly identifies the side (right or left) of the image where the joint space narrowing occurs with high accuracy as determined by visual inspection of images.

Local Variance. The outputs of this method highlight the important areas of the pictures green while keeping the image in it's original state. Same analysis as for guided backpropagation for differentiating the correct side of the image applies. From visual inspection the method worked well both for high and low severity classes. Note however that images of class 0 the joint space will be highlighted just as for high severity. In both cases its size determines the prediction. From our observation the method wasn't able to pick up on more sophisticated features of the images, like cysts for example.

Front-end

Formative research. The interview with the physician yielded the following list of requirements:

(1) A metric for the prediction certainty.
(2) Faster XAI methods (at that point only integrated gradients was implemented, our slowest XAI method).
(3) Alternatively, provide the physicians with the option to obtain the KL prediction with no XAI image (to shorten the waiting time).
(4) Documentation to guide physicians in the choice of model, XAI, and how to interpret the results.
(5) All input data should be stored locally for patient privacy.

Pre-validation interface. We refer the reader to Figures 14, 15, 16 and 17 in the Appendix for a visual representation of the Recurrent Care Application. In addition, a demo of our interface can be found at the following link: **https://youtu.be/uGal-7geSow**. Following the advice of the interviewed physician and the time constraints, the team was able to implement all points from the list above except point 4). As a result, the Recurrent Care App is made of three pages. The **"Home"** page (cf., Figure 14) is designed to provide the user with a short tutorial on how to use the app. In particular, it describes the content of each of the pages. Next, the **"Upload your Data"** (cf., Figure 15) page enables to doctor to query an image by patient number or to upload an image from their personal files. Finally, the input images are opened on the **"Prediction"** page. On this page the physician can find an HITL component: our drawing tool to draw circles, rectangles, or free shapes around the areas that they consider important for the KL prediction. In addition, the physician can input what they believe the KL score is, and they can write a short diagnostic text. On the same page, the physician is able to select a model from the following list: DenseNet201, ResNet152V2, InceptionV3, and Xception. In addition, the physician can select an XAI method of their preference from the following selection: Integrated Gradients, LIME, Guided Backpropagation, and Guided Backpropagation + Local Variance (cf., Figure 16) . All XAI methods can be combined will all our models. Alternatively, if the doctor is only interested in the predicted KL score and not the XAI method, they may choose "None" as an XAI method. The model then runs, and returns (if selected) the corresponding XAI image, as well as

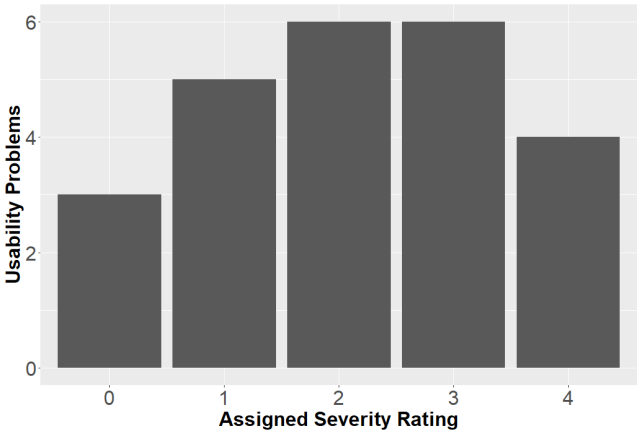

Figure 13: Distribution of UX problem severity ratings

a KL severity prediction, and the certainty of the model. The physician may also choose to display the severity grading graph, which shows the certainty of each KL class. Finally, the physician has the option to download the XAI images) (cf., Figure 16).

User Validation

Think Aloud Protocol. The Think Aloud sessions brought to light 24 usability problems. The distribution of the severity can be seen in Figure 13. These problems were classed into the categories found in Table 5.

Class	Description	Number of UX problems
Information flow	The previous data is not saved when changing pages, refreshing the page, or adding input to a page.	4
Communication	The user is not updated with respect to the current state of the app. The use of language is not correct.	7
Visibility	The features of the app are not clearly visible.	4
Efficiency	The app is slow.	1
Functionality	The app does not provide all the features that the user needs.	4
Documentation	The documentation available to the user is insufficient or incorrect.	4

Table 5: Usability problem classes and instances

Four UX design problems were rated with a 4: a usability catastrophe. Three of these were related to the information flow and they were related to the workflow of Streamlit: the script runs from the start every time the page is refreshed or updated with new user input. Hence, the previous user input is deleted and the information based on that input is no longer displayed (e.g., output from the selected XAI method). These correspond to problems 21, 22 and 23 on Table 7 in the Appendix.

The fourth catastrophic usability problem fell under the class "Documentation": there was no documentation explaining physician what each model and XAI method did (see problem 24 in Table 7). The rest of the usability problems can be found in Table 7 . Nonetheless, the majority of the usability problems were considered minor, cosmetic, or not applicable, and the experience of the users was still enjoyable. In fact, several of the expert's comments praised the clarity of the application. In addition, all users were able to complete all the tasks, yielding a completion rate of 100% .

SUS score. The mean SUS score was approximately 79.5, with the following 95% confidence interval [63.1, 95.9] . The distribution of the scores can be found in Table 3.

Question	Median (First Quantile, Third Quantile)
1	4 (3,4)
2	2 (1,2)
3	4 (4,5)
4	2 (1,2)
5	3 (3,4)
6	1 (1,2)
7	5 (5,5)
8	2 (1,2)
9	4 (4,4)
10	1 (1,2)

Table 6: Distribution of the scores per question of the SUS questionnaire (see Table 3)

Post-validation interface

Following the Think Aloud and SUS questionnaire results, we solved three of the four usability problems with severity 4. These correspond to problems 21, 22 and 24 on Table 7. In addition, we solved problems 20, 15, 12, 11, 10, 9, 4, 2 and 1 in order of priority given their severity score ad the time

constraints. This constitutes half of the usability problems found in the Think Aloud session. The solution to these issues or the reasons why the remainder of the usability problems could not be solved can be found on Table 7. See Figures 18, 19 and 20 in the appendix for a visual representation of the post-validation interface.

5 DISCUSSION

In this paper we introduced a system that includes several different convolutional neural network architectures as well as XAI methods, including a novel method improving existing state-of-the-art XAI approaches. The prediction results as well as the visualizations of the explanation methods were visualized in front-end built with streamlit. One major goal of this system was not to replace a doctor, but to provide an additional support system to further enhance the trust between patients and doctors. Therefore, we provided several options for practitioners to interact with the system as a human-in-the-loop. Finally, we found a way to systematically evaluate our application and received high scores. Still, our system suffers from various shortcomings. Similar to previous approaches [9], it was not possible achieve appropriate results for KL-class 0 and 1. The same applies to some XAI methods which were successful in finding relevant features for severe cases of osteoarthritis but only partly successful in identifying relevant parts for a healthy knee. This points to one of the major weaknesses of our system. It is lacking an evaluation method for the XAI methods. We tried to implement automated XAI evaluation methods [39] which did not fit the time constraints; hence, we performed manual user validation.

User Validation

Think Aloud. We worked with a team of five medical informatics experts to evaluate the Recurrent Care App. Physicians are the intended user group of this application; thus, we recognise the limitations of not collaborating with them during our Think Aloud sessions. Consequently, we cannot accurately estimate how likely it is that a physician would trust the system and use it in practice. Nonetheless, user validation can be seen as a iterative strategy. The first stages of an application may be completely different from the final product. Medical Informaticians are confronted daily with the need to design products aimed at health care professionals; hence, they are in tune with physician's needs as well as with the technical aspects of machine learning. We believe that collaborating with medical informatics experts helped us bridge the gap between AI and health sciences. The usability problems that surfaced during the Think Aloud sessions enabled us to further improve our interface. We believe that a second round of user validation with the collaboration of physicians would likely be more successful after these improvements. In particular, certain technical aspects related to the lack of user documentation in our pre-validation interface may have previously discouraged physicians. Five users is the recommended number of participants in the Think Aloud protocol since it often reveals over 80% of the usability problems [19]. This is particularly visible by the long list of usability issues presented in Table 7; where the most urgent issues have been addressed in the post-validation interface. Hence, we believe the application is currently more suitable and more likely to be appreciated by physicians during a second round of user validation.

SUS score. The SUS test we conducted is one of the most popular usability tests [40].The SUS score of the pre-validation Recurrent Care application is equal to 79.5, meaning that this app is on the 79th percentile of apps in terms of its usability. This translates to a good (borderline excellent) application. Nonetheless, the SUS questionnaire was only filled by the experts who participated in the Think Aloud sessions; hence, the sample size for this usability method is only five persons (which is the recommended number for the Think Aloud method, but not necessarily for the SUS method). This may explain the large the 95% confidence interval [63.1, 95.9] that we obtained. With this level of uncertainty, no conclusions can be drawn regarding the usability of the application.

We can however have a look a the individual SUS scores per question (Table 6). The odd numbered questions refer to positive aspects of the interface; hence, numbers close to 5 are desirable (Table 3). Conversely, the even numbered questions refer to negative aspects of the interface; hence, numbers close to 1 are desirable. The Recurrent Care app lost most usability points in question 5 with a median score of 3. This question relates to proper integration of the functions in the interface; which seems to be in line with the findings of the Think Aloud sessions (see Table 5 and Table 2): four usability problems were related to the information

flow in the application, and three of these were rated with a severity of 4. Conversely, all experts agreed on question 7 which relates to the expected ease-of-use of the application from the perspective of physicians. The Recurrent Care app had a median score of 5 in this question, which is in line with completion rate of 100% that was achieved by the Think Aloud participants. Globally, from this first round of user validation we concluded that the integration of the functionalities of the Recurrent Care application needed further improvement. Nonetheless, we also verified that we were on the right track towards our objective: creating an application with a simple physician-oriented design.

Post-validation interface

The post-validation interface is an improved product based on the insight from the Think Aloud sessions and the SUS questionnaire results. In particular, our pre-validation interface suffered from poor function integration. We fixed two from the four usability problems related to the Information Flow in the interface (problems 21 and 22 on Table 7). Nonetheless, problems 18 and 23 remain unsolved. These issues are related to Streamlit's work mode. The script runs from scratch every time a page is reloaded or updated. This implies that any input from the user, or output based on the input from the user disappears when new input is added. While we were able to solve half of the problems related to this issue by manually saving data in session states, the challenge remains. In fact, Streamlit is not originally designed for apps with multiple pages. We were aiming for a seamless design that used Python from the back-end to the front-end; however, our front-end solution is not currently suitable for commercial use. One could imagine using the Recurrent Care App for clinical studies, especially given its HITL components. Nonetheless, before that becomes a reality, it would be necessary to re-implement the application with a modern web development method such as React. On the other hand, Problem 24 was related to the lack of documentation provided to doctors. In the results section of the Front-end > Formative research, we listed five key requirements that the physician we interviewed deemed necessary for the suitability of this application for medical practice. In our pre-validation interface we had not integrated point 4), a usability mistake that was spotted by the medical informatics experts during the Think Aloud sessions. We rated this issue with a severity score of 4 since our aim was to create an app with a physician-oriented design. Giving this usability problem priority, we added information buttons underneath the model and XAI method menus to provide user support for the physicians.

These issues account for three out of the four severity score-4 usability problems that urgently needed to be solved. While not all usability issues were addressed, Table 7 offers a summary of the problems that were solved, the problems that were not solved with a reason or an idea of a solution, and the problems that are not solvable. Additionally, Figures 18, 19 and 20 can be compared to Figures 14, 15, 16 and 17 to observe how these changes translate visually. As a result, the post-validation Recurrent Care app offers a product with satisfactory usability that requires further fine-tuning of the integration of its functions, and that is tailored to the needs of physicians.

6 FUTURE WORK

More often health care is driven towards Value-Based Care (VBC). This model of care is based on the principle that health should be patient-centered, focused on medical conditions rather than single diseases, and span the entire cycle of care [41]. Future work could focus on integrating this osteoarthritis severity prediction model with other disease prediction models. This would facilitate the diagnosis of the comorbidities a patient may be experiencing and guide the doctor towards the best line of care for that patient. In addition, while the Recurrent Care application serves as a second opinion for the doctor, it could be taken beyond this role. Integrating this application with the rest of the electronic patient record could reveal its potential as a clinical decision support system. This would require applying the current guidelines for the treatment of osteoarthritis [42]. As a result, the application would not only return the degree of osteoarthritis, but also a recommendation regarding the best treatment for a patient with the given conditions; e.g., weight loss, surgical replacement of the knee, etc. Offering patients treatment following the gold standard of care for their specific conditions would be without a doubt a step towards VBC and better quality of life. In order to be able implement such a system it would be essential to resolve the issues mentioned in the discussion.

REFERENCES

[1] D. of Economic and S. Affairs, "Ageing and disability," 2022.

[2] C. for Disease Control and Protection, "Prevalence of doctor-diagnosed arthritis and arthritis-attributable activity limitation — united states, 2010–2012," 2012.

[3] O. A. Alliance, "Oa prevalence and burden," 2019.

[4] N. Arden, F. Blanco, C. Cooper, A. Guermazi, D. Hayashi, D. Hunter, M. K. Javaid, F. Rannou, F. Roemer, and J.-Y. Reginster, *Atlas of osteoarthritis*. Springer, 2014.

[5] J. W. Bijlsma, F. Berenbaum, and F. P. Lafeber, "Osteoarthritis: an update with relevance for clinical practice," *The Lancet*, vol. 377, no. 9783, pp. 2115–2126, 2011.

[6] J. Kellgren and J. Lawrence, "Radiological assessment of osteoarthrosis," *Annals of the Rheumatic Diseases*, vol. 16, no. 4, p. 494–502, 1957.

[7] D. L. Riddle, W. A. Jiranek, and J. R. Hull, "Validity and reliability of radiographic knee osteoarthritis measures by arthroplasty surgeons," *Orthopedics*, vol. 36, no. 1, pp. e25–e32, 2013.

[8] S. Kessler, K. Guenther, and W. Puhl, "Scoring prevalence and severity in gonarthritis: the suitability of the kellgren & lawrence scale," *Clinical rheumatology*, vol. 17, no. 3, pp. 205–209, 1998.

[9] K. A. Thomas, Ł. Kidziński, E. Halilaj, S. L. Fleming, G. R. Venkataraman, E. H. Oei, G. E. Gold, and S. L. Delp, "Automated classification of radiographic knee osteoarthritis severity using deep neural networks," *Radiology: Artificial Intelligence*, vol. 2, no. 2, p. e190065, 2020.

[10] E. LaRosa and D. Danks, "Impacts on trust of healthcare ai," in *Proceedings of the 2018 AAAI/ACM Conference on AI, Ethics, and Society*, pp. 210–215, 2018.

[11] D. M. E. T. M. J. J. Lior Shamir, Nikita Orlov and I. G. Goldberg, "Wndchrm – an open source utility for biological image analysis," 2008.

[12] N. E. O. K. M. Joseph Antony, Kevin McGuinness, "Quantifying radiographic knee osteoarthritis severity using deep convolutional neural networks," 2016.

[13] E. R. P. L. Aleksei Tiulpin, Jérôme Thevenot and S. Saarakkala, "Automatic knee osteoarthritis diagnosis from plain radiographs: A deep learning-based approach," 2018.

[14] N. D. F. Mark D. Kohn, Adam A. Sassoon, "Kellgren-lawrence classification of osteoarthritis," 2016.

[15] S. Mohseni, N. Zarei, and E. D. Ragan, "A multidisciplinary survey and framework for design and evaluation of explainable ai systems," *ACM Transactions on Interactive Intelligent Systems (TiiS)*, vol. 11, no. 3-4, pp. 1–45, 2021.

[16] K. G. G. M. A. V. Bas H.M. van der Velden, Hugo J. Kuijf, "Explainable artificial intelligence (xai) in deep learning-based medical image analysis," 2021.

[17] A. Holzinger, "Interactive machine learning for health informatics: when do we need the human-in-the-loop?," *Brain Informatics*, vol. 3, no. 2, pp. 119–131, 2016.

[18] N. N. Group, "Thinking aloud: The 1 usability tool," 2012.

[19] K. M., "3. thinking aloud eye tracking (11min)." University Lecture, 2022.

[20] P. Chen, "Knee osteoarthritis severity grading dataset," Sep 2018.

[21] O. Initiative, "Osteoarthritis initiative study description," 2020.

[22] C. Shorten and T. M. Khoshgoftaar, "A survey on image data augmentation for deep learning," *Journal of big data*, vol. 6, no. 1, pp. 1–48, 2019.

[23] Z. Zhong, L. Zheng, G. Kang, S. Li, and Y. Yang, "Random erasing data augmentation," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 34, pp. 13001–13008, 2020.

[24] Z. Hussain, F. Gimenez, D. Yi, and D. Rubin, "Differential data augmentation techniques for medical imaging classification tasks," in *AMIA annual symposium proceedings*, vol. 2017, p. 979, American Medical Informatics Association, 2017.

[25] S. R. J. S. Kaiming He, Xiangyu Zhang, "Densely connected convolutional networks," 2017.

[26] L. v. d. M. K. Q. W. Gao Huang, Zhuang Liu, "Deep residual learning for image recognition," 2015.

[27] Y. J. P. S. S. R. D. A. D. E. V. V. A. R. Christian Szegedy, Wei Liu, "Going deeper with convolutions," 2014.

[28] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," 2017.

[29] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A Large-Scale Hierarchical Image Database," in *CVPR09*, 2009.

[30] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *The journal of machine learning research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[31] T. B. M. R. Jost Tobias Springenberg, Alexey Dosovitskiy, "Striving for simplicity: The all convolutional net," 2015.

[32] M. Sundararajan, A. Taly, and Q. Yan, "Axiomatic attribution for deep networks," in *International conference on machine learning*, pp. 3319–3328, PMLR, 2017.

[33] M. T. Ribeiro, S. Singh, and C. Guestrin, """ why should i trust you?" explaining the predictions of any classifier," in *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pp. 1135–1144, 2016.

[34] Streamlit, "The fastest way to build and share data apps," 2022.

[35] S. J. Adams, R. D. Henderson, X. Yi, and P. Babyn, "Artificial intelligence solutions for analysis of x-ray images," *Canadian Association of Radiologists Journal*, vol. 72, no. 1, pp. 60–72, 2021.

[36] A. Bangor, P. T. Kortum, and J. T. Miller, "An empirical evaluation of the system usability scale," *Intl. Journal of Human–Computer Interaction*, vol. 24, no. 6, pp. 574–594, 2008.

[37] S. J., "Rating the severity of usability problems," 2013.

[38] S. A., "The system usability scale and how it's used in ux," 2020.

[39] Y.-S. Lin, W.-C. Lee, and Z. B. Celik, *What Do You See? Evaluation of Explainable Artificial Intelligence (XAI) Interpretability through Neural Backdoors*, p. 1027–1035. New York, NY, USA: Association for Computing Machinery, 2021.

[40] K. Finstad, "The usability metric for user experience," *Interacting with Computers*, vol. 22, no. 5, pp. 323–327, 2010.

[41] M. E. Porter and E. O. Teisberg, *Redefining health care: creating value-based competition on results*. Harvard business press, 2006.

[42] S. L. Kolasinski, T. Neogi, M. C. Hochberg, C. Oatis, G. Guyatt, J. Block, L. Callahan, C. Copenhaver, C. Dodge,

D. Felson, *et al.*, "2019 american college of rheumatology/arthritis foundation guideline for the management of osteoarthritis of the hand, hip, and knee," *Arthritis & Rheumatology*, vol. 72, no. 2, pp. 220–233, 2020.

APPENDIX

Pre-Validation Interface

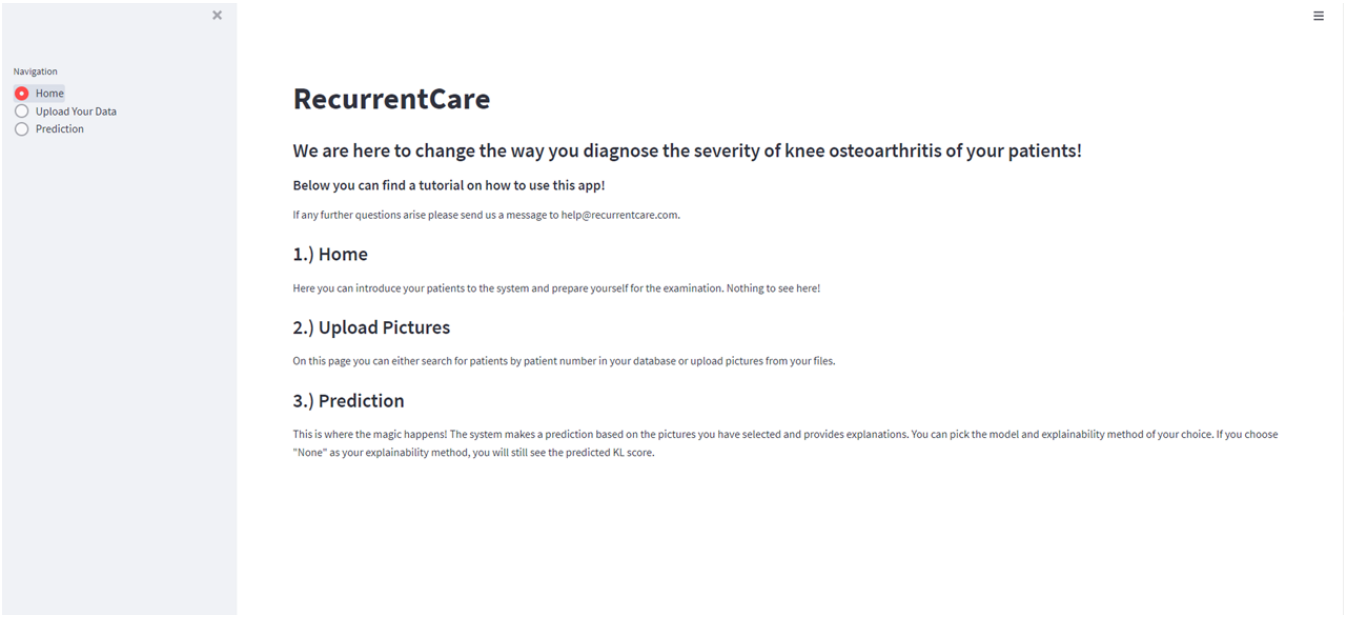

Figure 14: "Home" page of the Recurrent Care Application. This page displays a short tutorial on how to use the Recurrent Care Application.

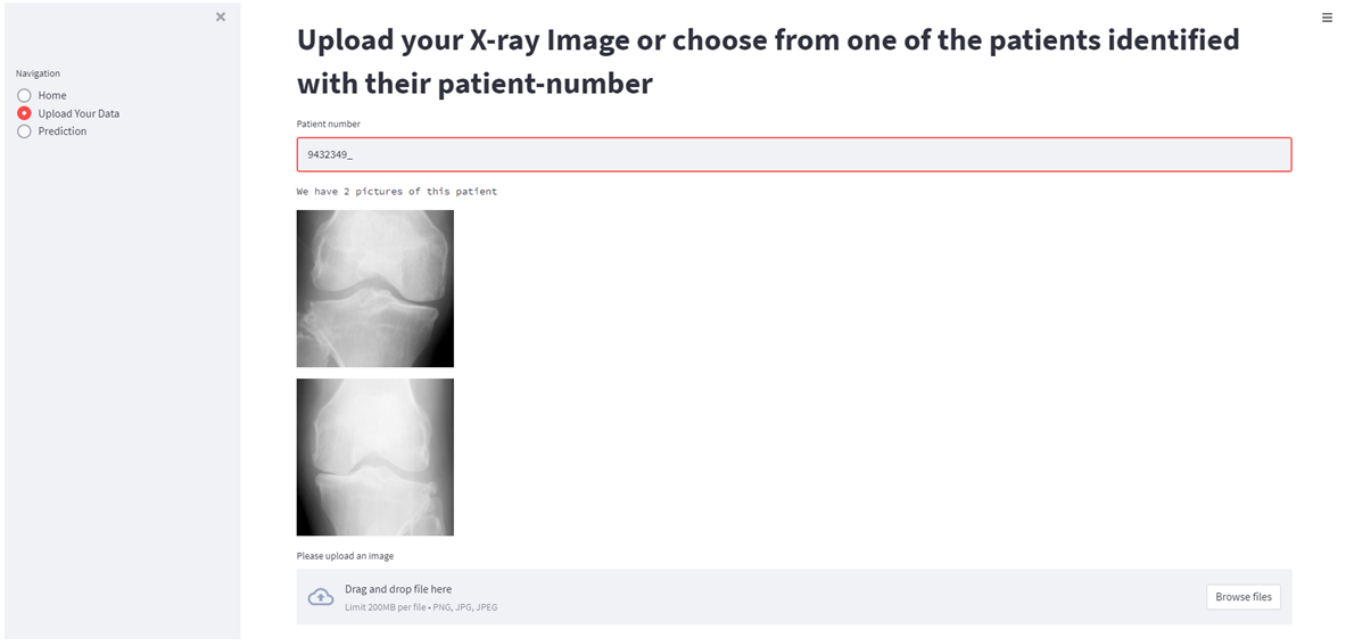

Figure 15: "Upload your Data" page of the Recurrent Care Application. In this case the images have been uploaded by typing in the patient number. On the bottom of the page, the button "Browse files" offers the possibility to upload images from local files.

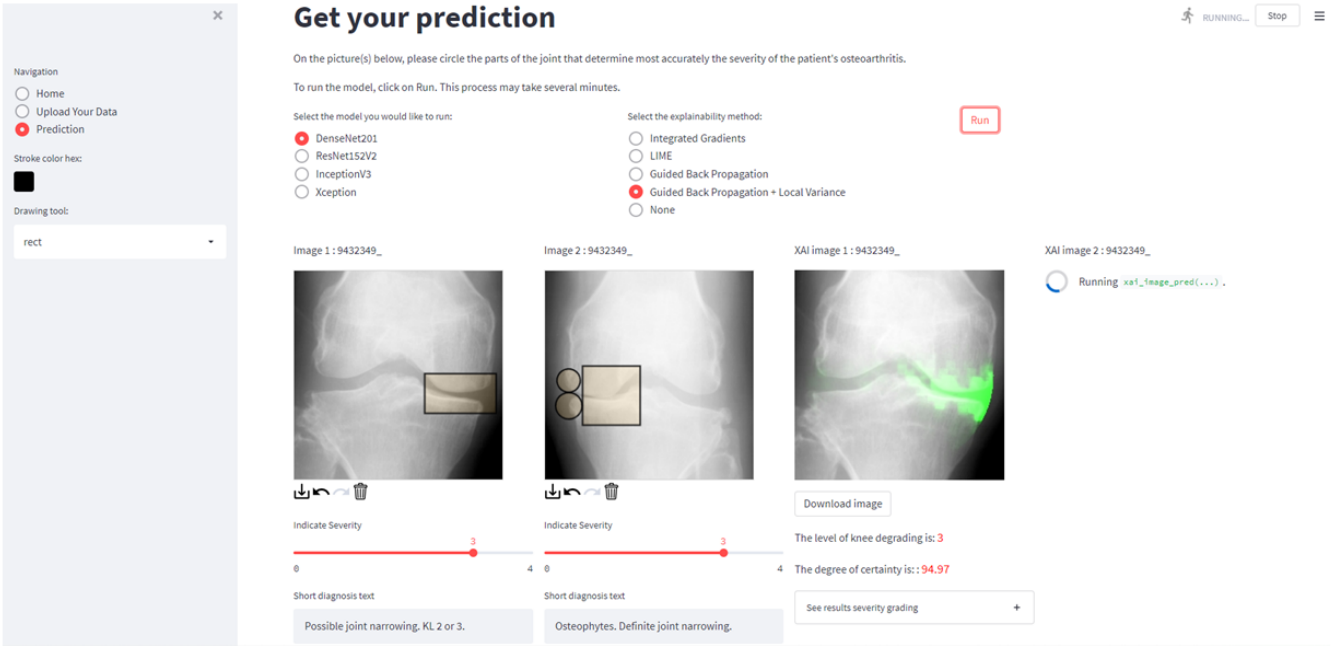

Figure 16: "Prediction" page of the Recurrent Care Application. This figure shows the top of the "Prediction" page while the xAI method is loading.The physician has indicated the most important areas for the prediction, a KL score and a short diagnostic text.

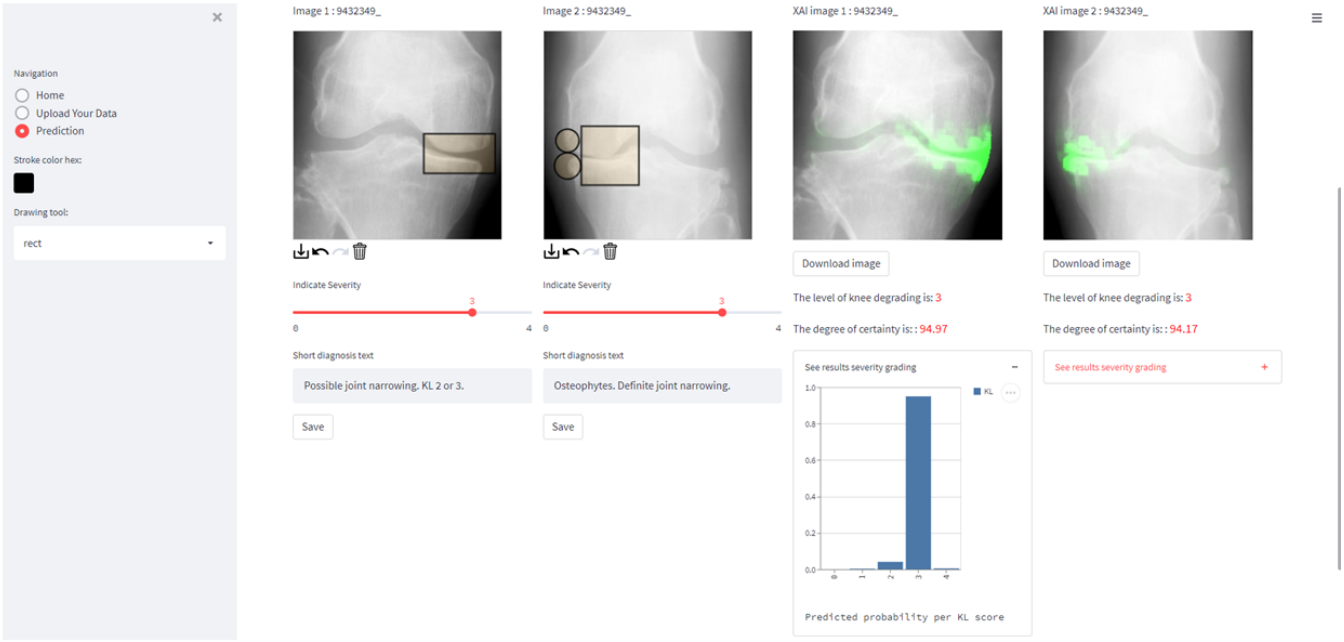

Figure 17: "Prediction" page of the Recurrent Care Application. This figure shows the bottom of the "Prediction" page after running the model. The xAI image is downloadable. The plot of the prediction certainty is shown when clicking on "See results severity grading".

Post-Validation Interface

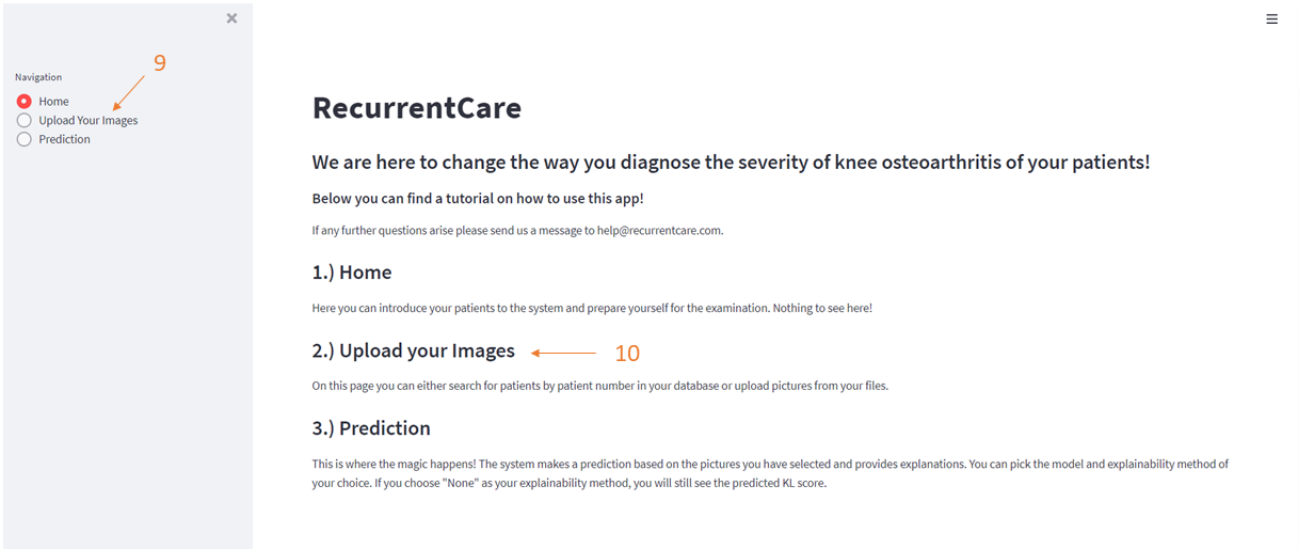

Figure 18: "Home" page of the Recurrent Care Application after validation. The arrows point at changes from the pre-validation app in Figure 14. The numbers refer to the usability problem tackled as labeled in Table 7.

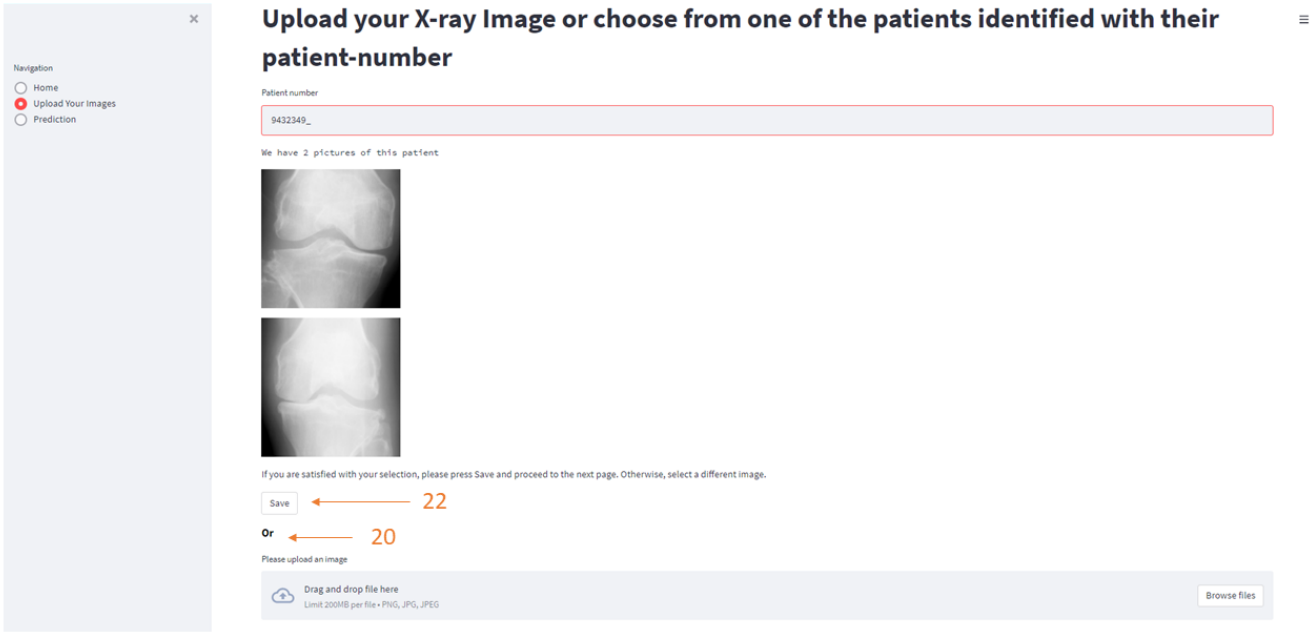

Figure 19: "Upload your Images" page of the Recurrent Care Application after validation. The arrows point at changes from the pre-validation app in Figure 15. The numbers refer to the usability problem tackled as labeled in Table 7.

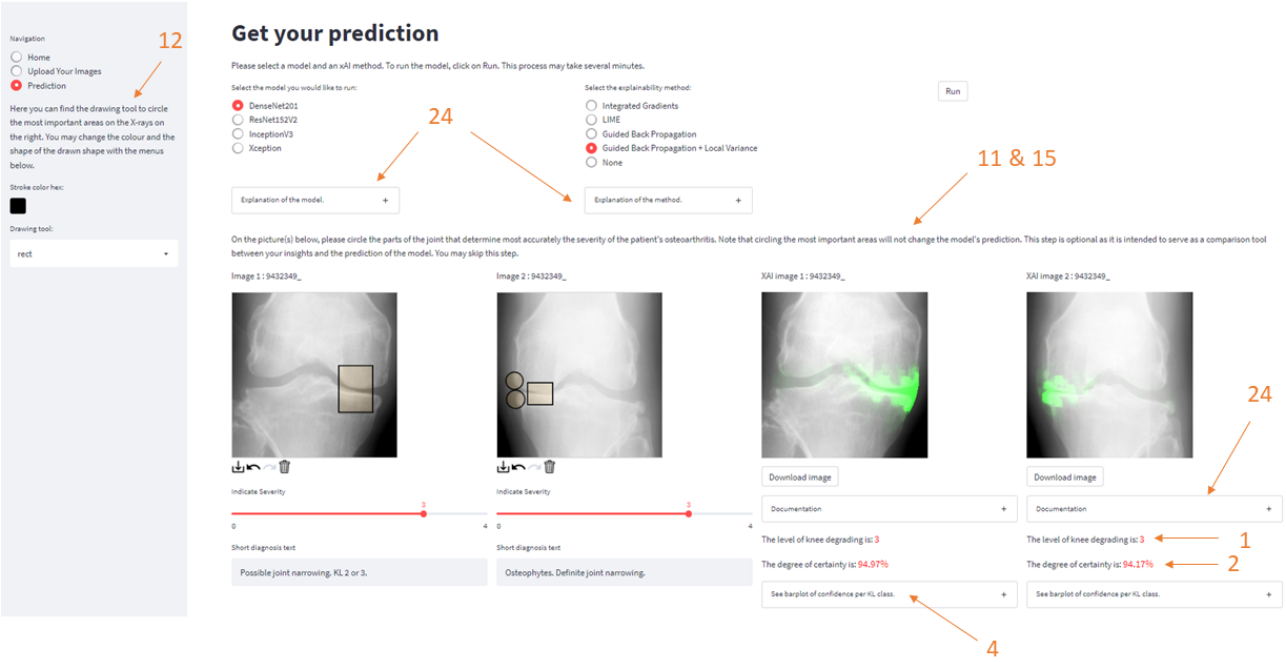

Figure 20: "Prediction" page of the Recurrent Care Application after validation. The arrows point at changes from the pre-validation app in Figures 16 and 17. The numbers refer to the usability problem tackled as labeled in Table 7.

Remark: The button **"Explanation of the model."** (underneath the model menu) displays the following text when clicked: "These options are prediction models. The best performing one is DenseNet201. The remaining models also have a high accuracy, we recommend using any of the other options after running DenseNet201. It will enable you to compare the results and agreement between the models, this way you get four second opinions!".
The button **"Explanation of the method."** (underneath the xAI menu) displays the following text when clicked: "These options are xAI methods, they offer different styles to show which areas of the X-ray the model concentrated on in order to predict the KL score. The purpose of the xAI methods is to prove that the model indeed looked at the relevant areas and did not make a guess at random. Selecting any of these methods is up to your own preference, it will not change the prediction, but only the style in which the important areas are presented. Note that if a part of the knee is displayed as an important area for the prediction it does not necessarily mean that the area shows any signs of pathologies. For instance, for a healthy knee the app will display the areas of the knee that indicate that this knee is indeed healthy. Similarly for a knee with a predicted KL score of 4, the app will display the areas where the knee seems to have to most severe signs of osteoarthritis. Thus, most likely, osteophytes, severe joint narrowing, etc. ".
The button **"Documentation"** (underneath the xAI image) displays the following text when clicked: "The green highlighted pixels on the image are crucial to the classification decision of the neural network. These highlighted pixels are correlated with the predicted severity grade.".

Table 7: Think Aloud Results - Usability problems: classification and severity The column "Solved?" refers to whether the problem was solved after user validation. N.S.: Not solvable

Index	Problem	Category	Severity	Solved?	Comment
1	Message with KL prediction ends with a typo "::".	Communication	0	Yes	-
2	Certainty percentage misses the "%" symbol.	Communication	0	Yes	-
3	Title of plot of prediction certainty per KL class not entirely visible.	Visibility	0	N.S.	Only an issue in small screens.
4	Name on expandable button of plot of prediction certainty per KL class is not representative of its content: plot is not easily found.	Visibility	1	Yes	Current name: "See barplot of confidence per KL class".
5	Position of models and xAI methods should appear underneath X-ray.	Visibility	1	No	Only mentioned by one expert at the Think Aloud session. Interesting for future work.
6	When models run, message shown is "xai_image_pred(...)".	Communication	1	N.S.	Shown automatically by Streamlit. Should show message in human language; e.g., "Model is running".
7	Drawing tool options show "rect" instead of "rectangle"	Communication	1	N.S.	Part of st.drawable_canvas design.
8	Interface does not label knees as left or right knees.	Functionality	1	No	Hard to implement. Doctors can easily tell from location of the fibula.
9	Non-representative title on "Upload your Data" page.	Communication	2	Yes	Current title: "Upload your Images".
10	Tutorial on "Home" page refers to "Upload your Data" page as "Upload pictures".	Documentation	2	Yes	Current tutorial refers to "Upload your Images" as "Upload your Images".

Index	Problem	Category	Severity	Solved?	Comment
11	Text mentioning that users can circle the most important areas on the X-ray is separated from the X-rays by model and xAI method menus.	Communication	2	Yes	Text currently shown under model and xAI method menus.
12	Drawing tool menu on the navigation bar is not visible.	Visibility	2	Yes	Added description and instructions on navigation bar.
13	Long running-time of xAI methods.	Efficiency	2	N.S.	Depends on the machine being used. Hospitals could get access to high-performance computing hardware.
14	Missing "download all" button for xAI images. Currently the doctors need to download the images individually.	Functionality	2	N.S.	Future work - did not fit time constraints.
15	Unclear if indicating important areas on the X-ray will influence the model's prediction. / Unclear if this step is optional.	Documentation	3	Yes	Added disclaimer to Prediction page.
16	No progress bar for model prediction and xAI output.	Communication	3	N.S.	Future work - did not fit time constraints.
17	When selecting images from local folder, only one image can be selected at a time but usually both knees are diagnosed simultaneously.	Functionality	3	No	Future work - did not fit time constraints.
18	Patient number and uploaded images in "Upload your Data" are not saved after going back and forth one page.	Information flow	3	No	Future work - did not fit time constraints. Can be fixed by saving session state.
19	No "next page" button. Browser arrows do not work. Must find navigation buttons.	Functionality (/Visibility)	3	No / N.S.	Streamlit is still developing functionality for MultiApp apps.
20	In "Upload your Data", unclear that there are two options to upload pictures. Most experts believed entering the patient number was for documentation purposes and that the image had to be selected from the local files.	Documentation	3	Yes	Added "or" to "Home" page to show the two options.

Index	Problem	Category	Severity	Solved?	Comment
21	Pages take too long to load. Consequences : scrolling down too soon in "Prediction" sends user back to "Home". Drawable canvas fades momentarily when scrolling down in "Prediction". When drawable canvas is loading, if input severity score, user is send back to "Home" and drawn figures are deleted.	Information flow	4	Yes	Saved session states to avoid data loss.
22	X-ray uploaded from local files is not successfully imported if other X-rays have previously been uploaded by patient number.	Information flow	4	Yes	Added "Save" button in "Upload your Images" page. Session set is reset to zero.
23	When clicking on download button of an xAI image, all predictions and xAI images disappear.	Information flow	4	No	Future work - did not fit time constraints. Issues: Streamlit reruns entire script.
24	No documentation explaining the purpose and interpretation of the models and xAI methods.	Documentation	4	Yes	Added information buttons under model and xAI method menus with documentation.