RAM
ROBOTICS
AND
MECHATRONICS

SYSTEMATICALLY ADDRESSING COMPUTED TOMOGRAPHY HEMORRHAGE CASES TO IMPROVE DIAGNOSTIC SKILLS: A COMPUTER ASSISTED LEARNING SYSTEM

N.E.D. (Nick) in het Veld

BSC ASSIGNMENT

Committee:
dr. ir. F. van der Heijden
E.I.S. Hofmeijer, MSc
dr. ir. B.J.F. van Beijnum

April, 2021

024RaM2021
Robotics and Mechatronics
EEMathCS
University of Twente
P.O. Box 217
7500 AE Enschede
The Netherlands

UNIVERSITY OF TWENTE. | TECHMED CENTRE
UNIVERSITY OF TWENTE. | DIGITAL SOCIETY INSTITUTE

Abstract

In this research CT images possibly containing intracranial hemorrhage are systematically ranked by exploiting the varying performance of 70 trained deep neural networks. Using the exploited information, a computer-assisted learning (CAL) system is developed, aimed at improving intracranial hemorrhage diagnostic skills in radiology trainees.

By exploiting the varying performance of the trained deep neural networks, possible intracranial hemorrhage cases in a test dataset are assigned a rank using item analysis. Using item analysis classification guidelines found in literature, the cases are respectively put into a level system. An application containing the CAL-system is developed in MATLAB, in which this level system is incorporated. To evaluate the effectiveness of the CAL-system between different groups, participants who respectively have either a medical background or a non-medical background have been recruited for this research to obtain user performance data, who are given a test before and after the CAL-training.

The accuracy of the participants with a non-medical background increased from 69.6% to 79.2% after CAL-training, showing a significant increase in diagnostic performance ($p = 0.0125$). However, the accuracy of the participants with a medical background stayed the same at 81.0% after CAL-training, showing no significant increase in diagnostic performance ($p > 0.05$).

Overall, the findings in this research suggest that the CAL-system developed in this research has potential for the training of freshmen radiology students and/or people with no medical background, to bring their intracranial hemorrhage diagnostic skills to a baseline level.

Abbreviations

ICH Intracranial hemorrhage

CT Computed tomography

DNN Deep neural network

CAL Computer-assisted learning

Contents

1 Introduction **3**
1.1 Problem assessment 4

2 Theory & Background **5**
2.1 Neural networks 5
2.2 Item analysis 6
2.3 Feedback-based learning 8
2.4 Computer-assisted learning 9

3 Method **10**
3.1 Analysis 10
3.2 Ranking scheme 12
3.3 Application 14
3.4 Participants 21

4 Results **23**

5 Discussion **26**
5.1 Major aspects 26
5.2 Minor aspects 30

6 Conclusion **33**

Bibliography **36**

Appendix **37**

Chapter 1

Introduction

Intracranial hemorrhage, also abbreviated as ICH, is a collective term compassing the extravascular accumulation of blood within different intracranial spaces [1]. Depending on the location of the ICH, the cause may vary. [2] However, often it occurs afters trauma with a blunt head injury. ICH encompasses four types of hemorrhages: epidural hemorrhage, subdural hemorrhage, subarachnoid hemorrhage, and intraparenchymal hemorrhage.

Likewise, symptoms may vary depending on the type of ICH [2]. Common symptoms, however, are headaches, nausea, vomiting, lethargy, seizures, neurological damage and/or decreased consciousness.

While other medical imaging techniques are used for the detection of ICH, a computed tomography (CT) scan is commonly performed. Acute blood is markedly hyperdense compared to brain parenchyma, thus a trained radiologist is able to diagnose the location of the ICH [1]. However, misidentifications of ICH do occur. The most common types of ICH that are missed are subdural and subarachnoid hemorrhages [3]. In the case of misdiagnosis of subarachnoid hemorrhages, the prognosis is severe, leading to poor clinical outcomes [4].

To aid the radiologist in the identification of ICH, researchers have explored the use of deep learning. [5] Here, convolutional neural networks are trained to identify ICH in CT images. While radiologists make the decision in the final diagnosis, these trained deep neural networks (DNN) can serve as a valuable "second opinion"-tool. Using this computer-assisted detection (CAD) tool, diagnosis time can be reduced and overlooked ICHs can be identified by radiologists.

In the past two decades, researchers have shown interest in and studied the use of computer-assisted learning (CAL) for medical students. A similar study has shown [6] that a CAL-system has high potential for the training of medical students and health professionals in interpreting chest radiographs.

1.1 Problem assessment

Radiologists are trained to diagnose diseases from medical images. Before having enough experience to do so, they follow a trajectory as a radiology trainee to build up diagnostic skills. However, radiology trainees are mostly trained on cases that are currently being assessed or treated, or on cases that the supervisor has available. This results in a varying degree of experience among graduated radiologists. Also in their future career, their diagnostic skills will be dependent on the cases they will encounter. To level out their diagnostic skills, they would benefit from a CAL-system to practice with a variety of cases. To address this problem, a solution is formulated in this research by creating a CAL-system, in which users can train their diagnostic skills in a systematic manner. To come to a solution, however, the following research questions need to be addressed:

- How can a framework for the CAL-system be achieved which maximizes the learning effectiveness?
- How can a ranking system be achieved in which cases are systematically ordered?

First, by making use of information concerning neural networks, feedback-based learning and computer-assisted learning, a framework is constructed aimed at maximizing the learning effectiveness of the to-be-developed CAL-system. Furthermore, using methods found in item analysis to analyze the quality of items, cases in this CAL-system will be systematically shown to a user in an orderly manner. Then, the performance of 70 trained DNNs for this research is analyzed using a test dataset, where information derived from this analysis is exploited to systematically give the cases in the test dataset a rank by making use of the methods found in item analysis.

- How can the CAL-system be realized?
- How effective is the developed CAL-system in improving ICH diagnostic skills?

By designing an application in MATLAB with a functioning front end and back end, the CAL-system is tested on participants. After that, the performance data from participants is analyzed to determine if the CAL-system has significantly increased the diagnostic performance in the participants or not.

- How can the developed CAL-system be used for the radiology field?

Finally, various aspects of this research are addressed, some of which requiring major attention and some of them requiring minor attention, if the CAL-system is to be further developed. At the end of this report, a statement is made if the developed CAL-system has potential in increasing ICH diagnostic performance in radiology trainees.

Chapter 2

Theory & Background

To understand how neural networks operate and what kind of information they can provide obtaining information from literature is necessary. Furthermore, information is required that can provide a framework to rank cases in a systematic way, for which information generated by the trained neural networks is exploited to eventually assign a rank to cases in a test dataset. In this research it is assumed that this ranking, using neural networks as the ’test-taking’ group, is able to gauge the difficulty of a case for humans. Lastly, it is of importance to understand which methods in the pedagogy are used to maximize the efficiency of learning, but also to understand which conditions need to be fulfilled to make a CAL-system highly effective.

2.1 Neural networks

[8] A neural network is obtained by a concatenation of several layers of neurons, also called perceptrons. A perceptron is a mathematical model of a biological neuron. Each perceptron has a certain weight and bias. The weights can be positive, negative, or zero. A positive weight encourages the perceptron to fire, while a negative weight inhibits it from doing so.

The first layer of a neural network corresponds to the input, while the last layer corresponds to the output. The inner layers are called hidden layers. These hidden layers contain transfer functions and logistic functions, which are deterministic functions of the inputs. If it is a network where the information flows forward from the input to the output, it is called a feedforward neural network, alternatively called a multi-layer perceptron (MLP). On the other hand, when images are used as inputs, the neural network is also considered a convolutional neural network (CNN).

The purpose of the units in the hidden layer in a CNN is to learn non-linear combinations of the original inputs; this is called feature extraction. Here, so called filters scan the input image for patterns. A CNN is effectively a MLP in which the hidden units have local receptive fields, and in which the weights are tied or shared across the image. Using these weights, the resulting network is able to exhibit translation invariance, meaning it can classify patterns no matter where they occur inside the input image. The DNNs which are trained for this research accept a CT image as input and outputs the classification of the image. This classification is then compared with the true label of the image to determine the accuracy of the DNN.

Besides the classification output, other types of information can be obtained from a DNN, such as occlusion sensitivity maps [9,10]. These are effectively heat maps which show which parts of an image are considered the most important by a DNN for its classification decision. Here, different portions of the input image are systematically occluded with a grey square, during which the output of the classifier is monitored. A deconvolutional neural network is used to map the feature activities in the intermediate layers back to the input image, to determine which patterns in the input image resulted in a specific activation in the feature maps. To do so, a deconvolutional neural network is attached to each of the layers of the convolutional neural network. By unpooling, rectifying and filtering these activations, the activity in the layer beneath which resulted in these specific activations is reconstructed.

Other methods exist that map feature activations back to the input image, such as GRAD-CAM [11] or LIME [12]. Overall, LIME is considered the simplest of the methods. On the other hand, a GRAD-CAM map usually has a lower spatial resolution than an occlusion map and can therefore miss finer details [11].

2.2 Item analysis

Item analysis is a method that is used to evaluate the effectiveness of an item in a test. It is primarily used in the pedagogy to evaluate the quality of an item in a test, for which items are kept, revised or discarded depending on their quality relative to the test. Here, two aspects of an item are analyzed: (i) the item difficulty and (ii) the item discrimination.

2.2.1 Item difficulty

The item difficulty, also called the P-value [1] or difficulty index, is a value that indicates the difficulty of an item in a test. This is equal to the proportion or percentage of test-takers who answered an item correctly [13]. Here, the larger the P-value, the easier the item is considered to be. It is noted, however, that the P-value is regarded as a behavioural measure, and it is not an intrinsic characteristic of an item. In this regard, the difficulty is defined in terms of the relative frequency of test-takers choosing the correct response.

In literature, various interpretations are found when it comes to classifying the difficulty of an item using the difficulty index, such as those found in the

[1] To avoid confusion with the p-value found in statistics, the capital letter P will be used in this paper to indicate the p-value associated with the item difficulty.

study of A. Bichi [14] or in the study of S. Marie et al [15], but no standardized classification scheme using the difficulty index is to be found. In literature, however, it is still noted that one of the guidelines is that items with a difficulty index lower than 0.20 or higher than 0.80 are to be discarded [16]. Nonetheless, a simple classification scheme for the difficulty index used in the study of E. Morales [17] and in the study of Hartati et al. [18] is the following:

Table 2.1: The classification scheme for the difficulty index used in the study of E. Morales [17] and Hartati et al. [18]

Difficulty classification	**Index value (P)**
Very Easy	0.81 - 1.00
Easy	0.61 - 0.80
Moderate	0.41 - 0.60
Difficult	0.21 - 0.40
Very Difficult	0.00 - 0.20

2.2.2 Item discrimination

On the other hand the item discrimination, also called the discrimination index, indicates the extent to which success on an item corresponds to success on the whole test [19]. In general, the discrimination index is computed by evaluating the difference in performance on an item between the high and low performing group on the test, where the degree of this difference is expressed in a value ranging from -1 to +1. To compute the standard discrimination index, it can be computed as a function of the number of correct answers on an item by the high and low performing group, and the size of each of these groups.

Another indicator that can be used to express the discrimination index of an item is the point-biserial correlation coefficient [13]. The point-biserial correlation is the Pearson correlation between responses to a particular item and scores on the total test, which shows how strongly these are interrelated, ranging from a value from -1 to +1. Here, the point-biserial correlation is used to find out if the right people are getting the items right. The advantage of using the point-biserial correlation coefficient over the standard discrimination index, is that every person taking the test is taken into consideration when computing the point-biserial correlation coefficient, while only approximately half of the test-takers is taken into consideration when computing the standard discrimination index.

Since all items in a test are intended to cooperate to generate an overall test score, any item with negative or zero discrimination undermines the test [13]. The higher the discrimination index, the better the item because such a value indicates that the item discriminates in favour of the upper group, which should get more items correct. On the other hand, items with poor discrimination require either major revision or should be eliminated from the test.

To calculate the point-biserial correlation coefficient in the case of item analysis, the following formula is used, which describes the relation between test-takers scoring an item (in)correct and their test score, while taking the standard deviation of the performance of the test-takers into account.

$$r = \frac{M_{correct} - M_{incorrect}}{\sigma} \cdot \sqrt{P_{correct} \cdot P_{incorrect}} \tag{2.1}$$

Here, the point-biserial correlation coefficient r is a function of: (i) the mean test score of the test-takers getting the item correct $M_{correct}$; (ii) the mean test score of the test-takers getting the item incorrect $M_{incorrect}$; (iii) the number of test-takers getting the item correct $P_{correct}$; (iv) the number of test-takers getting the item incorrect $P_{incorrect}$; (v) the standard deviation of the test scores across the test-takers σ. Using the found point-biserial correlation coefficient values for the items, the items can then be classified based on their value. The Ebel and Frisbie's guidelines for classifying item discrimination [16] is found in the table below:

Table 2.2: The classification scheme for the discrimination index by Ebel & Frisbie [16]

Discrimination classification	**Index value (r)**
Poor	< 0.10
Low	0.10 - 0.19
Acceptable	0.20 - 0.29
Good	0.30 - 0.39
Excellent	> 0.40

2.3 Feedback-based learning

Feedback is generally considered an effective tool for teaching and learning. Baydal et al. [20] evaluated the effect of computer-based immediate feedback on medical students' learning in a pharmacology course. The study found that immediate feedback had a positive impact on the students' self-directed learning, although it did not improve test scores.

In the study of Fazio et al. [21], data showed that only providing right/wrong feedback is not considered effective, but being able to review material is on the other hand an effective way of providing feedback. However, it was noted that the usefulness of right/wrong feedback might depend upon the nature of the to-be-learned material. Eventually, a suggestion was made that if feedback is to be provided, the feedback should give information about the correct answer, as opposed to simply marking a response as correct or incorrect without feedback.

An other study of Butler et al. [22] showed that, in case of trying to understand the material, explanation feedback resulted in better learning performance than correct/incorrect answer feedback. However, they noted that studies have shown no significant benefits when significantly increasing the complexity of the feedback message.

The study of Chamberland et al. [23] on the other hand suggested that providing the correct diagnosis and a simple content feedback improves the subsequent ability to correctly diagnose similar cases. Here, the study made a case that this could develop into an effective design of educational activities that use self-explanation to support the development of students' diagnostic reasoning. The study concluded that adding simple corrective feedback in the form of the correct diagnosis and making sure that students have the opportunity to process it, seemed to be a very simple measure to specifically improve the students' diagnostic ability for similar cases.

For the diagnosis of ICH in particular, the study of Watanabe et al. [24] has shown that after making use of computer-assisted detection (CAD), the diagnostic performance to correctly diagnose possible ICH cases, improved in all physicians who participated in the study.

2.4 Computer-assisted learning

Researchers have explored the use of using computer-assisted learning (CAL) in medical education. The use of a CAL-system enhances medical education and provides learning opportunities that cannot be taught by traditional methods [25].

Advantages of CAL in medical education are considered to be: (i) the computer provides the student with unlimited time, (ii) the computer is not judgemental, (iii) the CAL can be repeated frequently without the computer being impatient.

On the other hand, disadvantages are considered to be: (i) the CAL-system does not substitute academics, (ii) group working is left out of the CAL-system, (iii) conventional education still plays an important role in training.

The use of computers for learning is considered more effective when [26]: (i) the student is in control of learning instead of the teacher, (ii) feedback is optimized, (iii) peer learning is optimized, (iv) there is a diversity of teaching strategies, (v) there are multiple opportunities for learning,

Chapter 3

Method

To understand how the accuracy of each DNN trained for this research compares to each other, an analysis is performed. The information obtained from this analysis is eventually exploited to systematically order the cases in the test dataset. Using the found item analysis classification schemes (Table 2.1 and 2.2), the cases of the test dataset are systematically ranked. To realise the CAL-system in a visual format, an application is designed that incorporates the CAL-system. To evaluate the effectiveness of the CAL-system, participants are recruited to obtain data.

3.1 Analysis

For this research, 70 DNNs have been trained in MATLAB for the classification of CT images possibly containing ICH, to classify them into either a hemorrhage case (present) or a non-hemorrhage case (absent). For the training of the DNNs, a training dataset has been used containing 2465 CT images. In MATLAB the command *trainingOptions* has been used to specify the training options for the DNNs, which are shown below:

- MaxEpochs: 50
- ValidationPatience: Inf
- ValidationFrequency: 20
- Verbose: True
- Shuffle, every-epoch
- MiniBatchSize: 32
- InitialLearnRate: 0.001

The accuracy performance of the DNNs has been evaluated on a test dataset containing 1633 CT images in which ICH is either present or absent. Here, the accuracy achieved by the DNNs over this test dataset varied, ranging from approximately 65% to 80%, which is visualized in Figure 3.1.

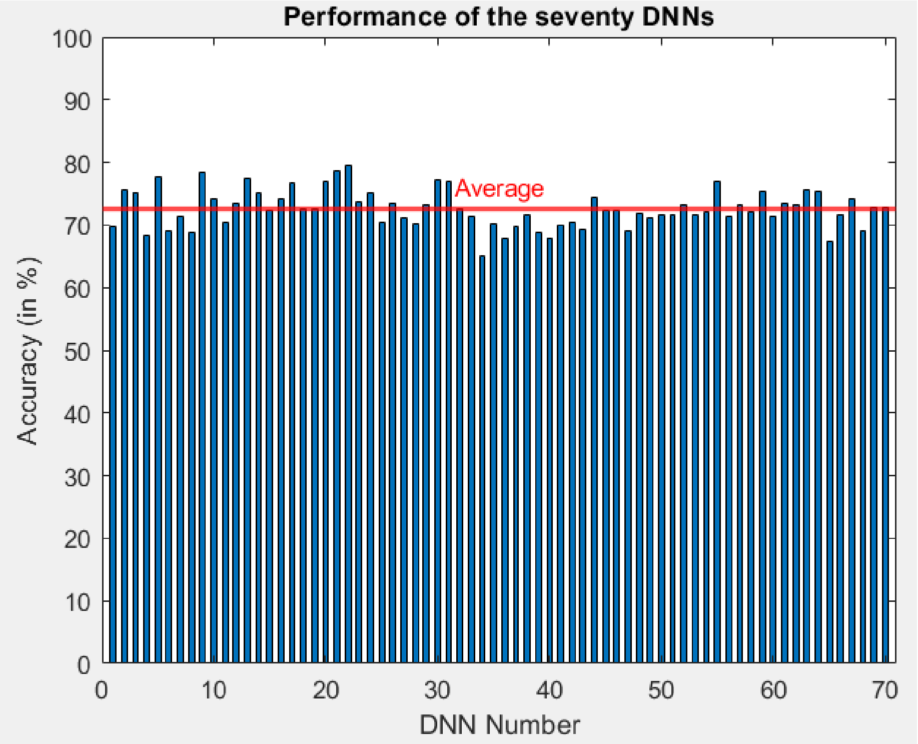


FIGURE 3.1: The accuracy scores over the test dataset across the 70 DNNs. The red line indicates the average accuracy.

The average accuracy of the 70 DNNs over the test dataset was found to be 72.61% ± 3.00%. By computing the correct-incorrect answer distribution of the 70 DNNs on a case, the difficulty index of a case is eventually found. Using this information, the found standard deviation of the test accuracy scores across the DNNs, and the test scores of the DNNs over the test dataset, the discrimination index of a case is computed, expressed in the point-biserial correlation coefficient (Equation 2.1). Using the found difficulty index and discrimination index of a case, the cases in the test dataset are eventually assigned a rank.

3.2 Ranking scheme

To classify the difficulty of a case in the test dataset, a classification scheme based on the scheme shown in Table 2.1 is used (Figure 3.2). This is chosen because no classification schemes are described in literature that were specifically applicable for this type of research, thus a simple classification scheme is used for the difficulty index. In this classification scheme, cases with a difficulty index equal to zero, meaning that none of the DNNs classified these cases right (and assuming the same response in humans), are discarded and assumed invalid.

While the item discrimination classification scheme found in literature, as shown in Table 2.2, consists of five classifications, a modified classification scheme is used, for which some classifications were merged together to obtain a total of two classifications (Figure 3.3). In this classification scheme, cases with a negative discrimination index are discarded. However, cases with a discrimination index equal to zero are kept, which are advised to be discarded according to literature. The reason for this decision is that cases with a zero discrimination index and a non-zero difficulty index are the easiest cases (in this case, every DNN diagnosed the case correctly), therefore choosing to include these in the CAL-system; this decision however is touched upon in the *Discussion*. All in all, this means that there are five difficulty segments, for which in each of these segments there are two discrimination segments, for a total of ten levels (Figure 3.4). Using this level system, the cases of the test dataset are systematically given a rank.

One of the reasons to specifically have ten levels instead of a higher amount, which could be done by using the original item discrimination classification scheme as shown in Table 2.2, is because the limited amount of time one could ask of a participant meant that the participant is significantly limited in mobility when moving up the ranks. In result, this would make the level system redundant when dealing with a high number of levels, since a participant would not be able to reach these levels due to the time limit. Adding to this, introducing a larger amount of levels would result in a level system in which certain levels consist of a very small number of cases, resulting in often repeating cases when solving cases in that particular level. In the worst case scenario, this would result in the user relying on their memory instead of their diagnostic skills. Lastly, in this research the item difficulty is considered to be more important than the item discrimination, therefore choosing to merge the discrimination index segments instead of the difficulty index segments.

The goal of this level system is that a user starts with low discriminating cases in a difficulty segment, since the low performing users should be able to get these cases right. If a user is performing well, the user moves up to the higher discriminating cases, while staying in the same difficulty segment. Once again, if the user performs well enough, the user moves to a higher difficulty segment and will solve low discriminating cases again.

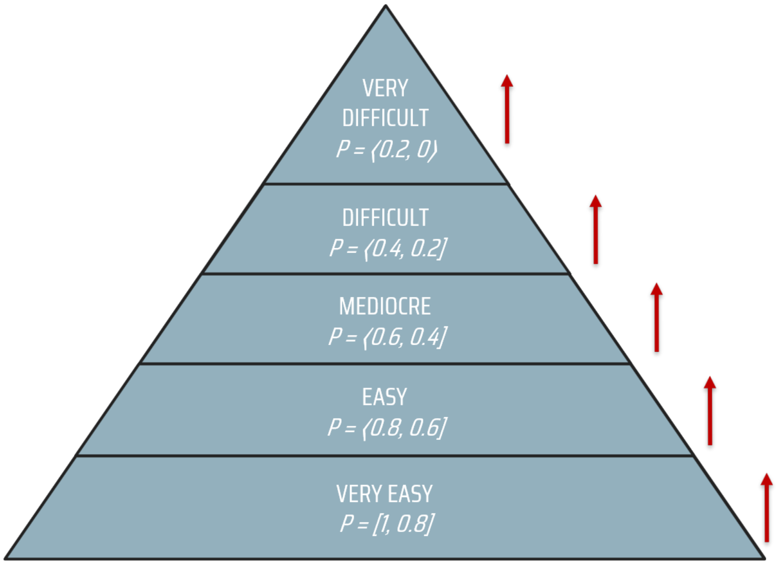


FIGURE 3.2: The difficulty index (P) classifications used for assigning ranks to cases in the test dataset, based on the classification scheme found in literature (Table 2.1). The difficulty segments are further separated in discrimination segments, as shown in Figure 3.3

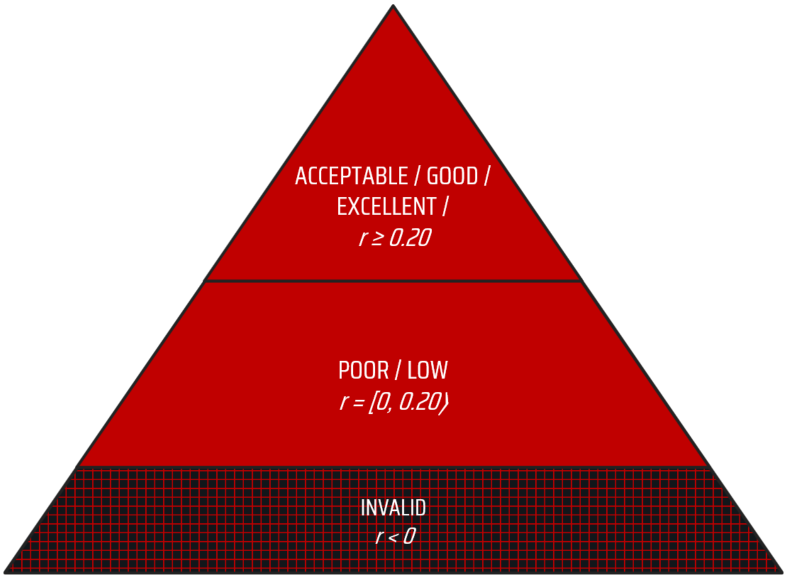
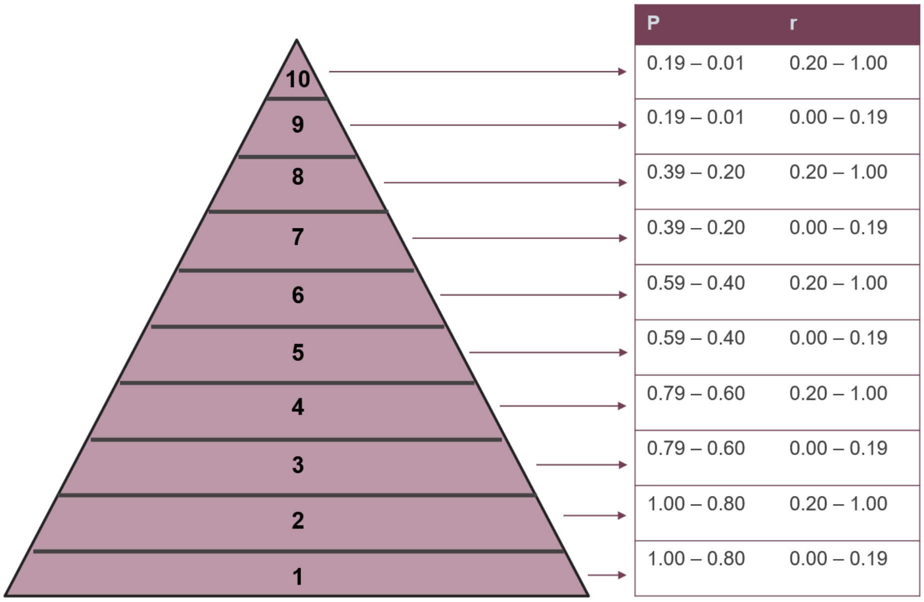


FIGURE 3.3: The discrimination index (r) classifications used for assigning ranks to cases in the test dataset, partially based on the classification found in literature (Table 2.2)



P	r
0.19 – 0.01	0.20 – 1.00
0.19 – 0.01	0.00 – 0.19
0.39 – 0.20	0.20 – 1.00
0.39 – 0.20	0.00 – 0.19
0.59 – 0.40	0.20 – 1.00
0.59 – 0.40	0.00 – 0.19
0.79 – 0.60	0.20 – 1.00
0.79 – 0.60	0.00 – 0.19
1.00 – 0.80	0.20 – 1.00
1.00 – 0.80	0.00 – 0.19

FIGURE 3.4: Global overview of the level system based on the item quality classifications. Here the difficulty index is denoted with P, the discrimination index (point-biserial correlation coefficient) is denoted with r.

3.3 Application

The application incorporating the CAL-system is developed in MATLAB using the *App Designer*-toolbox and *Stateflow*-toolbox. In *App Designer* the front end is primarily created. On the other hand, in *Stateflow* the back end is primarily created.

3.3.1 Front end

As mentioned before, the front end of the application is developed using the *App Designer*-toolbox in MATLAB. This allows a quick prototype of an application to be created if one is already familiar with MATLAB, without having to invest time in learning programming languages used for application framework design such as Python or C#.

When the user starts the application on their computer, the user is eventually greeted with the start screen (Figure 3.5). Here, the user has the option to obtain information about the global trajectory of the application in the info page. On the other hand, the user can start a tutorial to understand the controls and workings of the application.

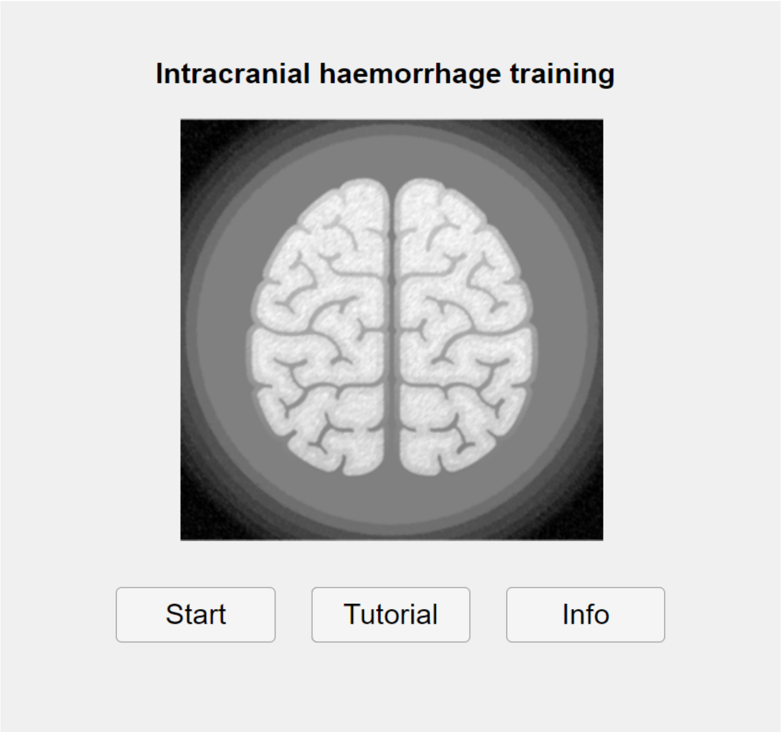
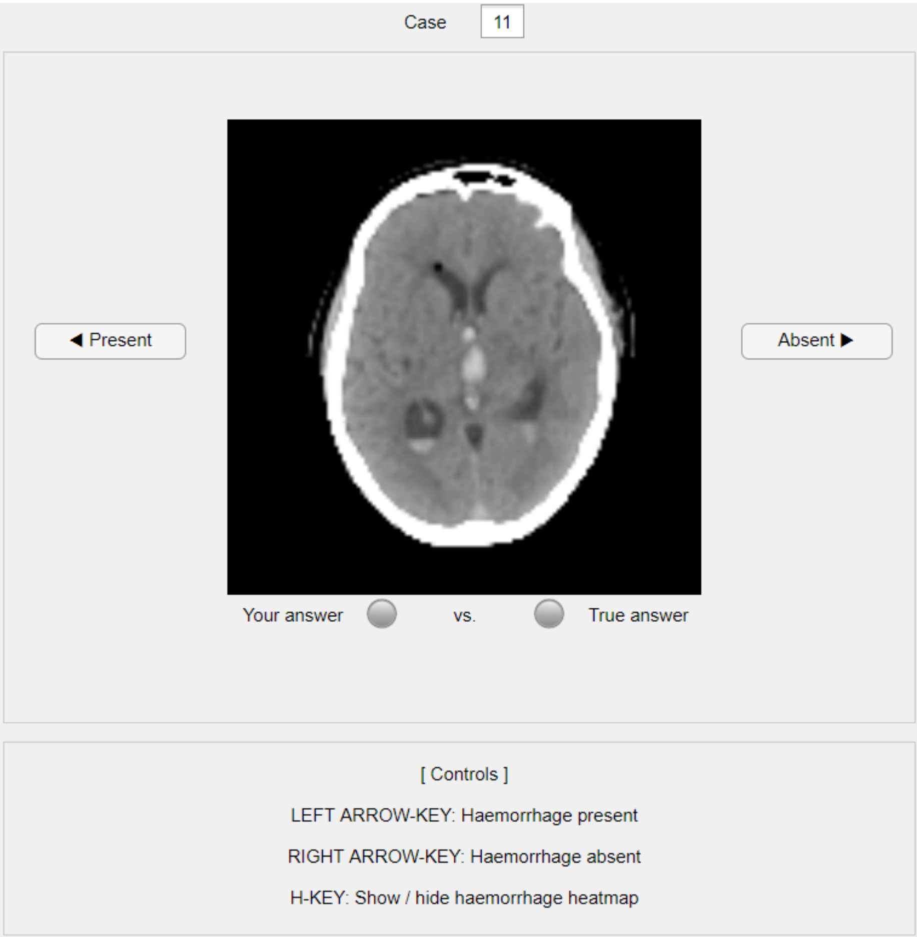


FIGURE 3.5: The start screen shown to the user in the application.

In the tutorial, the controls of the application are explained. Here, it is explained to the user is that cases can be solved by either making use of the mouse or by using keyboard inputs. With the left arrow-key, the user can answer that hemorrhage is present in the CT image. Likewise, with the right arrow-key the user can answer that hemorrhage is absent in the CT image. The main screen that is presented to the user in the application is shown in Figure 3.6 and 3.7.

After the user solves a case during the CAL-training, feedback is visually provided with color lamps (showing the user's answer and the correct answer) as well as a ”correct/incorrect answer” text-prompt (indicating if the user has diagnosed the case correctly), which is shown in Figure 3.7. Furthermore, the user is able to view an occlusion sensitivity heat map of the case with the H-key (Figure 3.8), serving as additional explanatory feedback. Finally, the spacebar-key is used to cycle through prompt messages and/or to load the following case when done reviewing a case during the training.

When the user is finished with the introductory part of the application, which comprises the info page and the tutorial, the user is able to start the CAL-system by pressing the START-button as shown in the start screen (Figure 3.5).



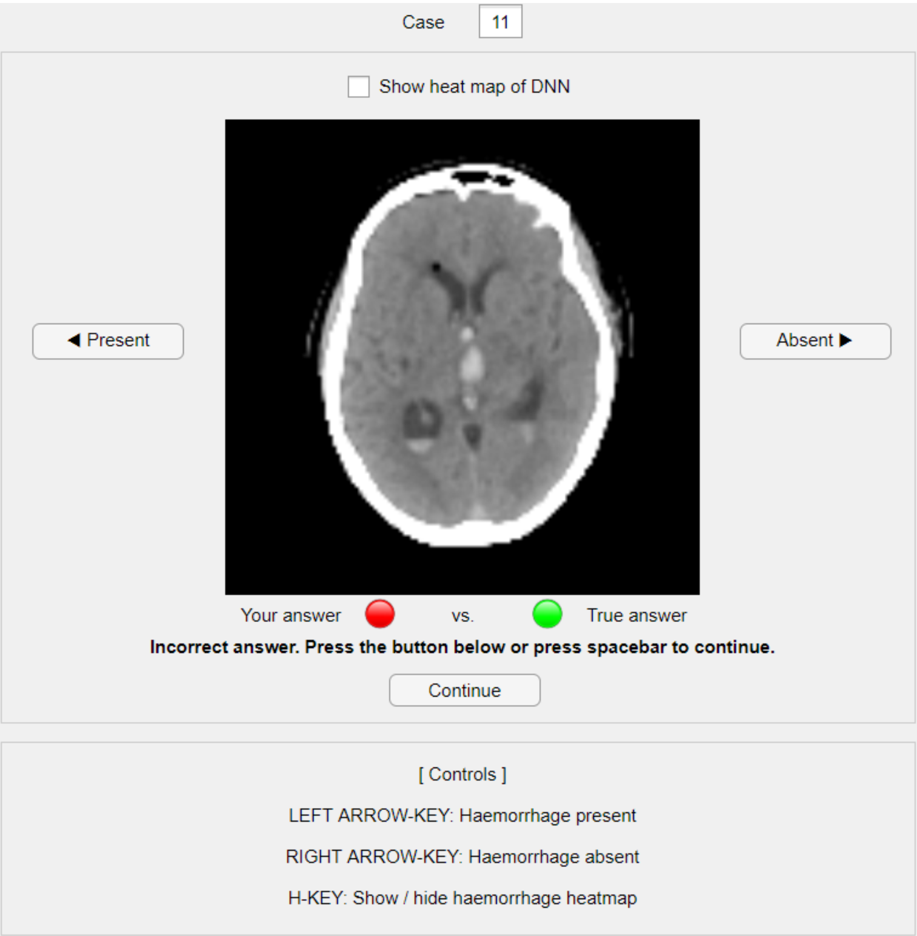
[Controls]

LEFT ARROW-KEY: Haemorrhage present

RIGHT ARROW-KEY: Haemorrhage absent

H-KEY: Show / hide haemorrhage heatmap

FIGURE 3.6: The main screen presented to the user during CAL-training. The user has not solved the case yet.



Incorrect answer. Press the button below or press spacebar to continue.

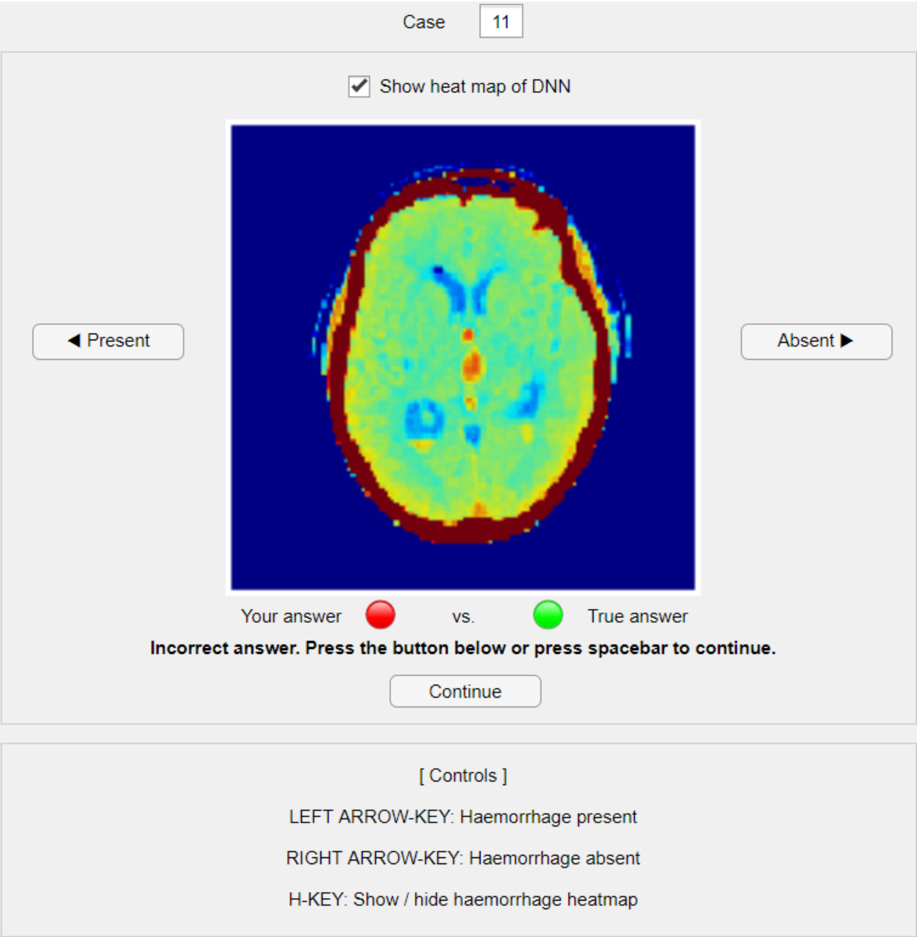
[Controls]

LEFT ARROW-KEY: Haemorrhage present

RIGHT ARROW-KEY: Haemorrhage absent

H-KEY: Show / hide haemorrhage heatmap

FIGURE 3.7: The main screen presented to the user during CAL-training. The user has solved the case and is able to review it. Feedback elements are shown to the user in the form of ”Your answer vs True answer” color lamps and a ’correct/incorrect answer’ text-prompt. A red color lamp is defined as ’the image does not contain ICH’, while a green color lamp is defined as ’the image does contain ICH’. Additionally, the user is able to view an occlusion sensitivity heat map of the case using the toggle-box above or alternatively by pressing the H-key (Figure 3.8).



Case 11

☑ Show heat map of DNN

Your answer 🔴 vs. 🟢 True answer

Incorrect answer. Press the button below or press spacebar to continue.

[Controls]

LEFT ARROW-KEY: Haemorrhage present

RIGHT ARROW-KEY: Haemorrhage absent

H-KEY: Show / hide haemorrhage heatmap

FIGURE 3.8: The main screen presented to the user during CAL-training. The occlusion sensitivity heat map of the case is shown, which is done by pressing either the checkbox above the image or the H-key on the keyboard.

3.3.2 Back end

Using the *Stateflow*-toolbox in MATLAB, a state machine is created to systematically develop logic for the application, but also to allow systematic transitions from one logic to the other to take place (Appendix A).

In the application, no neural networks are actively running, and all the images shown in the application are therefore pre-processed. The reason for this is because utilizing these neural networks as a part of the application requires heavy computational power, which in result requires a lot of system resources, making the application slow and unresponsive in the end.

From a global overview, the trajectory of the application consists of three phases (Figure 3.9): (i) a pre-training test, (ii) the CAL-training itself, and (iii) a post-training test. This scheme was chosen to be able to evaluate the effectiveness of the CAL-training (and the CAL-system overall), by allowing a user to go in blindly for the initial test, and being presumably more knowledgeable and skilled for the final test, after the user has gone through the CAL-training. In this scheme, depending on the user performance in the pre-training test, the user is placed in a suitable level when starting the CAL-training. Therefore, the pre-training test is also defined as a part of the CAL-system, besides the CAL-training.

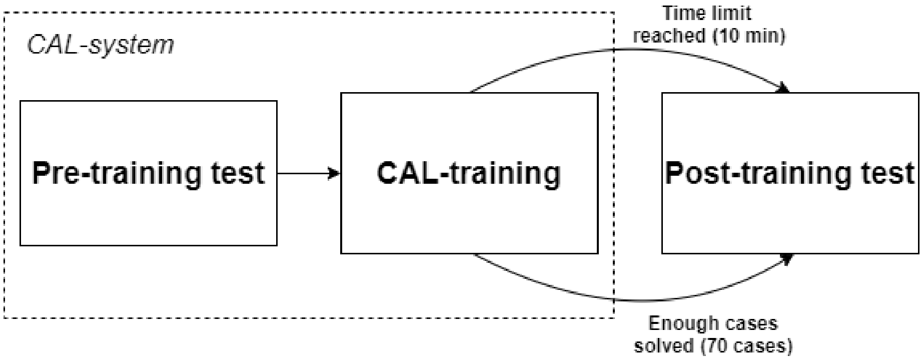


FIGURE 3.9: The trajectory of the application designed in MATLAB. The user finishes the CAL-training after solving 70 cases or if the timer passes the 10 minute mark. The CAL-system is defined as the pre-training test plus the CAL-training, and is explained and visualized in further detail in Figure 3.10

The pre- and post-training test consists of the same 40 cases. For this, two cases of each level were randomly picked in which ICH is absent. Similarly, two cases of each level were randomly picked in which ICH is present. These cases are not present in and/or shown during the training, to avoid that these cases are solved based on memory in the case these are included and shown during the training. In short, the test contains 20 cases in which ICH is present and 20 cases in which ICH is absent, to eventually obtain a test of mediocre difficulty (for which the DNNs had an average accuracy of 50%, see Figure 4.1) and with an equal-sized classification distribution.

The reason for creating a test for which the DNNs have an average accuracy score of 50%, is because it is unknown if the perceived difficulty by DNNs translates one-to-one into the perceived difficulty by humans. Therefore, to err on the side of caution and to avoid making the test too easy or too difficult, a test is constructed for which the average accuracy achieved by the DNNs on this test is approximately 50%.

On the other hand, the reason for choosing 20 cases each is to be able to evaluate the effectiveness of the CAL-system in improving the sensitivity and/or specificity of a user (besides the accuracy) in an equal fashion, while having a sufficient amount of cases to do so, but also by keeping in mind that the test should not consist of a large amount of cases due to the time constraint.

During the tests no feedback is provided. Here, feedback elements such as the "Your answer vs. True answer" color lamps, the correct/incorrect answer text-prompt and the occlusion sensitivity heat map toggle-box (Figure 3.7) are not shown and are removed from the main screen. Furthermore, no timer is set for the tests, as it is desired to have the same amount of cases to be solved for the pre-training test as for the post-training test. Besides, while the same 40 cases are shown to the user in these tests, the order in which these cases are shown is determined randomly, resulting in incomparable results when the test is early terminated due to an imposed time limit. After having finished the pre-training test, the user starts with the CAL-training and, depending on their pre-training test accuracy score (Table 3.1), is placed in a level.

Table 3.1: The user's accuracy in the pre-training test is used to place the user in a suitable level for the CAL-training. If the user's accuracy is perfect, the user is finished with the application, as training is considered redundant for these users.

Accuracy score	**Placed in level ...**
0.00 - 0.59	1
0.60 - 0.69	2
0.70 - 0.79	3
0.80 - 0.89	4
0.90 - 0.99	5
1	Exit

During the CAL-training, the user is able to move up or move down in the level system in a sink or swim fashion (Figure 3.10). After the user solves 10 randomly selected cases of a level, the accuracy of the user is checked. Depending on the value of this accuracy, the user is promoted or demoted to a higher or lower level respectively. If the user's accuracy, after solving these 10 cases, is above or equal to 80%, the user is placed a level higher; if it is lower than 80%, the user is placed a level lower. Here, the threshold value was chosen to be 80%, as it was observed in the study of Watanabe et al. [24] that physicians had an overall accuracy of approximately 80% when solving ICH cases. Therefore, it is desired to observe the same accuracy in a user when the user is finished with a level. During the training, feedback is provided in the form of correct/incorrect answer feedback and, as previously mentioned, in the form of occlusion sensitivity heat maps. When the user completes the training, the user starts with the post-training test (Figure 3.10).

Due to the concern for the limited time availability of a participant, and having no feedback about the total time a participant would require to go through the application (in specifically for the tests, since no timer is set here), the maximum amount of time a user was allowed to train themselves was set to 10 minutes (Figure 3.9). Here, a timer would start when starting the training and would stop when it passed the 10 minute mark, eventually stopping the training. Alternatively, the user was able to complete the training after solving 70 cases (Figure 3.9).

The reason for limiting the amount of cases for the training, in this case 70 cases, was because: (i) this allowed a user to have approximately 8.5 seconds, which is considered to be a lenient amount of time given, to solve and review a case; (ii) this chosen amount of cases allowed the user to have enough upwards mobility to reach a significantly higher level, while taking some leeway of level demotion into consideration; (iii) this avoided a really well performing user to only solve cases in level 10 for the rest of the duration of the training, since no solution was developed for users who keep succeeding in level 10; (iv) it took into account that the user would possibly experience some "solving fatigue" from their side, which would affect the training effectiveness and possibly the results in the final test.

3.4 Participants

To understand if the CAL-system is effective or not for people of varying experience and backgrounds, it is desired to find participants which have either a medical background or a non-medical background to ultimately obtain two homogeneous groups: participants with (i) a medical background or (ii) a non-medical background. By doing so, a statement can be made about the effectiveness of the CAL-system on these subgroups. To evaluate if the post-training performance scores of the participants are significantly higher than their pre-training performance scores, a paired one-tailed t-test (upper tail) will be used. Lastly, the level mobility of the participants will be analyzed to observe if the participants are able to move up the ranks or not, and to observe if the participants are able to reach a higher level than their begin level. The participants will be recruited based on convenience sampling [27], for which word-of-mouth is used as the recruitment method.

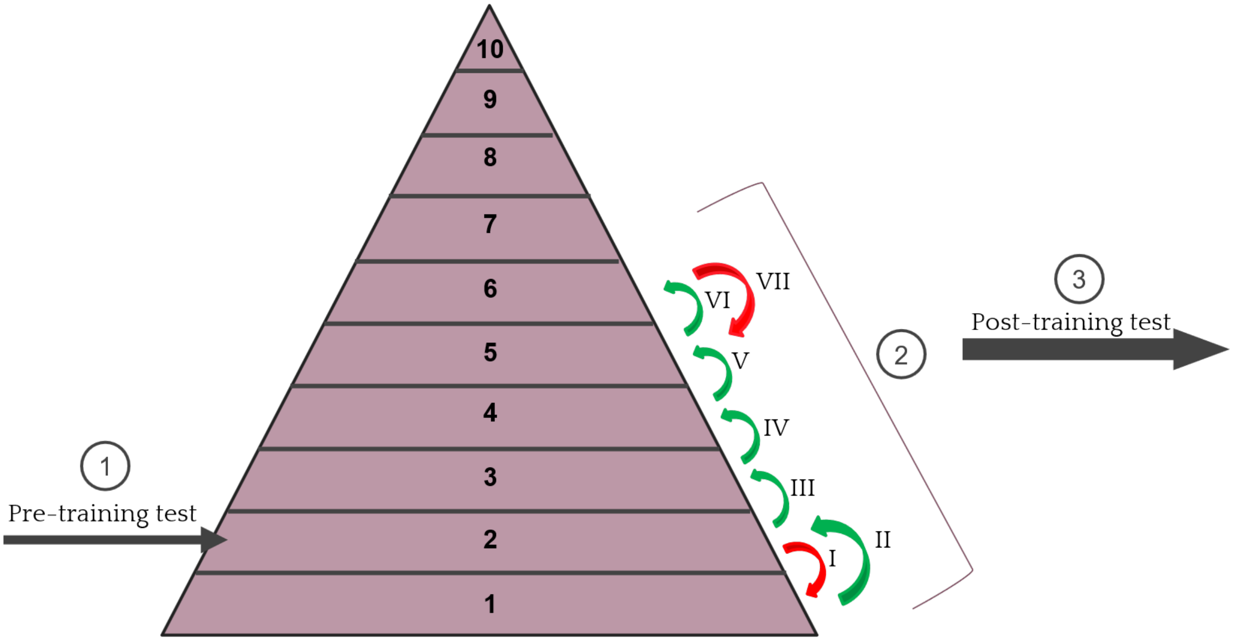


FIGURE 3.10: An example of the trajectory of a user starting and exiting the CAL-system: **(1)** The user achieved an accuracy score in the range of 0.60 - 0.69 in the pre-training test, and is placed in level 2 in the level system as described in Table 3.1; **(2)** The order in which the user moves up (green arrow) and/or moves down (red arrow) the levels is shown with the roman numerals I-VII. In each level the user is required to solve 10 randomly selected cases of that particular level, moving up or moving down a level depending on their accuracy after finishing the level (80% accuracy threshold); **(3)** The user has solved 70 cases, reaching an end level of 5, and completes the CAL-training.

Chapter 4

Results

For the evaluation of the effectiveness of the CAL-system, eight participants were found, some of them having a medical background and some of them not, These participants are ordered in their respective subgroups as described in Method: Participants. The medical participants consists of people pursuing a degree related to medicine or the health sciences, while the non-medical participants consists of people who are pursuing a study or have currently an occupation in the engineering, social sciences or business field. The average diagnostic performance of the participants and of their respective subgroups, before and after CAL-training, is shown in Table 4.1.

Table 4.1: The (difference in) test performance among the participants, before and after the CAL-training, expressed in percentages. A number of participants were found (n = 8): with a medical background (n = 3), and with a non-medical background (n = 5). The significance *p* of the results are evaluated with a paired one-tailed t-test (upper tail).

	b/ CAL (%)	a/ CAL (%)	Difference (%)	Significance (p)
All participants (n = 8)				
Accuracy	73.9	79.9	+ 6.0	0.08
Sensitivity	73.8	82.5	+ 8.7	0.09
Specificity	73.8	76.9	+ 3.1	0.34
Medical (n = 3)				
Accuracy	81.0	81.0	+ 0.0	0.50
Sensitivity	86.7	83.3	- 3.4	0.65
Specificity	75.0	78.3	+ 3.3	0.43
Non-medical (n = 5)				
Accuracy	69.6	79.2	+ 9.6	0.0125
Sensitivity	66.0	82.0	+ 16.0	0.04
Specificity	73.0	76.0	+ 3.0	0.37

a/, after; b/, before; CAL, computer-assisted learning

When taking a quick glance at Table 4.1, the numbers suggest that the overall diagnostic performance of the participants has increased; still, these results are not significant ($p > 0.05$). However, when decomposing the results, there is a significant difference to be observed between the two subgroups. The overall diagnostic performance of the participants with a medical background did not (significantly) improve. On the other hand, the participants with a non-medical background significantly improved their diagnostic performance, in particular their accuracy ($p = 0.0125$) and sensitivity ($p = 0.04$); however, no significant increase is observed in the specificity ($p > 0.05$). The average reading time of the participants before CAL-training, as well as for the medical and non-medical subgroups, was observed to be 3.6 sec. After CAL-training, the average reading time of the participants stayed in the same order of magnitude: 2.2 sec for all participants, 2.1 sec for the medical subgroup and 2.3 sec for the non-medical subgroup.

To gain a better understanding of how the test accuracy scores of the various (sub)groups, including the DNNs, compare to each other, a visual comparison of the average test accuracy scores of these (sub)groups is shown in Figure 4.1:

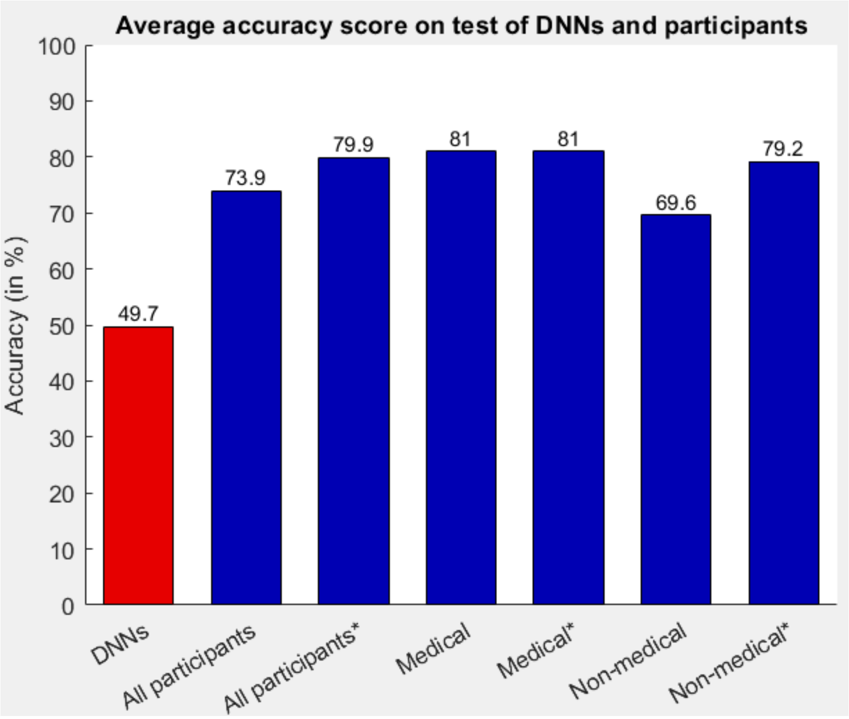


FIGURE 4.1: A visual comparison of the average accuracy score on the test between various (sub)groups. The average accuracy score of the DNNs is denoted in red. The asterisk sign * denotes that the (sub)group went through CAL-training.

Noteworthy is that a significant difference in the average test accuracy scores can be observed between the DNNs and the participants, in all cases the latter achieving higher accuracy scores than the former. Nevertheless, it can be observed that the accuracy score of the non-medical participants, after going through the CAL-system, approximates the post-training accuracy score of their medical counterpart (non-medical 79.2% vs. medical 81.0%).

Lastly, the level mobility (begin level, end level, peak level) of the participants is found in Table 4.2. Here it can be observed that the medical participants reached both a higher end and peak level with respect to their begin level (P1 - P3); for four out of five non-medical participants, similar results can be observed (P4 - P7). However, one out of five non-medical participants did not reach a higher end level with respect to their begin level, and their end level was even observed to be lower than their begin level (P8). Nonetheless, it can still be observed that this participant reached a higher peak level with respect to their begin level.

Table 4.2: The levels reached by the participants during the training. The begin level, end level and highest reached level (peak level) of the participants are shown in the columns. The participants are labeled with a number P#.

	Begin level	End level	Peak level
Medical (n = 3)			
P1	3	6	6
P2	4	7	7
P3	4	5	7
Non-medical (n = 5)			
P4	3	4	5
P5	2	7	7
P6	2	5	5
P7	3	4	5
P8	3	2	5

Chapter 5

Discussion

In the discussion, aspects which were overlooked in this research, aspects which posed limitations on this research, and other aspects concerning the CAL-system, are discussed in detail. These aspects are categorized into two categories: (i) aspects that require major attention and (ii) aspects that require minor attention.

5.1 Major aspects

5.1.1 User level mobility and performance

It may have been the case that allowing a participant to solve only 70 cases during their training is on the low side and should therefore have been increased to a higher amount, since it was observed that during the training the participants solved the cases in a quick pace. Furthermore, the CAL-system should have stopped when the user succeeded in level 10, or should have looped back to the participant to level 1 after succeeding in level 10, to fully make use of the maximum time set for the training, which was equal to 10 minutes. By doing so, a more effective training may have been realized, which would have positively affected the results.

A second point to be made is that, while the medical participants solved cases in and ended with a higher level, a direct relation between reaching a higher level and achieving a higher accuracy is unclear. Their non-medical counterpart on the other reached a higher end and peak level with respect to their begin level, while scoring significantly higher in their post-training test than their pre-training test. Overall, this may suggest that either the used level system on the whole is arbitrary, or that the level scheme (Figure 3.4) is only effective for people with no existing experience. In future research, therefore, various level systems must be tested on experienced and inexperienced users, to determine if different level systems must be designed for these groups.

Lastly, it is noteworthy that in Figure 4.1 there is a significant difference observable between the DNNs performance and that of the participants. This suggests that the perceived difficulty by the DNNs does not fully translate into perceived difficulty by humans.

5.1.2 Distribution of cases across levels

The level system that was used for the CAL-system, as shown in Figure 3.4, resulted in the following distribution in ICH-present and ICH-absent cases, shown in Figure 5.1 and 5.2. Here it can be observed that each level has an uneven distribution in ICH-present and ICH-absent cases. The inequality of the case distribution possibly has a significant negative influence on the effectiveness of the training, since it may be the case that a user figures out that they are more likely to classify a case correctly when continually choosing only one of either options. Therefore, in future research one should look at the distribution and should make the distribution in ICH-present and ICH-absent cases more equal to maximize the randomness of the cases, e.g. by leaving out a number of cases with a certain classification in a level.

When looking at Figure 5.1, it can be observed that level 1 contains a significant high amount of ICH-absent cases relative to ICH-present cases. This is due to all the cases with a discrimination index equal to zero are only to be found in level 1, which comprises cases where all 70 DNNs diagnosed the cases correctly. While the decision had been made to still include these cases in the CAL-system, it is advised in future research that cases with a discrimination index equal to zero are left out of the CAL-system, to avoid redundancy of very simple cases and to minimize the unequal distribution in a level.

5.1.3 Case retrieval in levels

When a user is solving cases in a level, a random case is chosen from a case-list associated with that level. However, when the user revisits the level during the training, it is possible that an already shown case may be presented to the user, since the same list is reloaded for that level. This in turn may negatively affect the outcome of the training, because there is a possibility that the user may rely on his memory instead of their diagnostic skills to solve a case. A possible solution is to implement a case "blacklist" to avoid a repetition of cases. However, this means that the CAL-system should have a sufficient high number of cases to be able to implement this, since in the extreme case the list would have to be reloaded again regardless, to avoid running out of cases to show in a level.

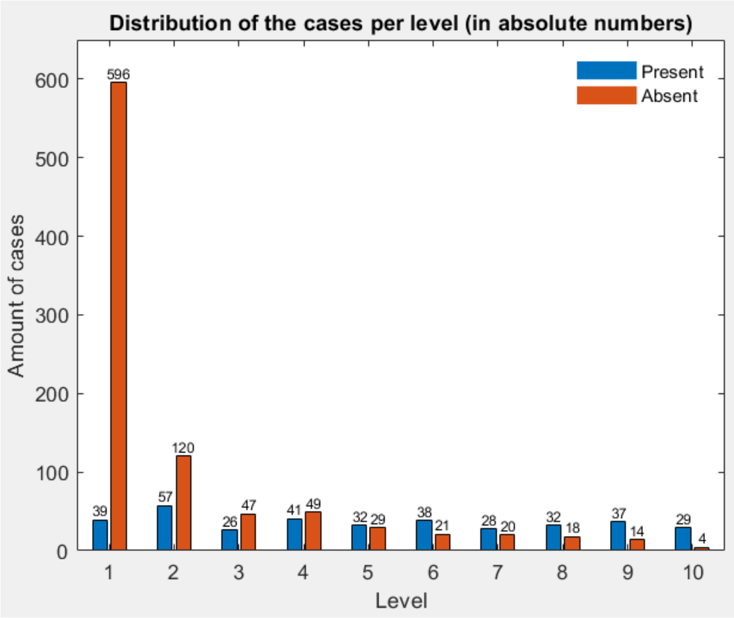


FIGURE 5.1: The distribution of the cases (present vs. absent) in each level, expressed in absolute numbers.

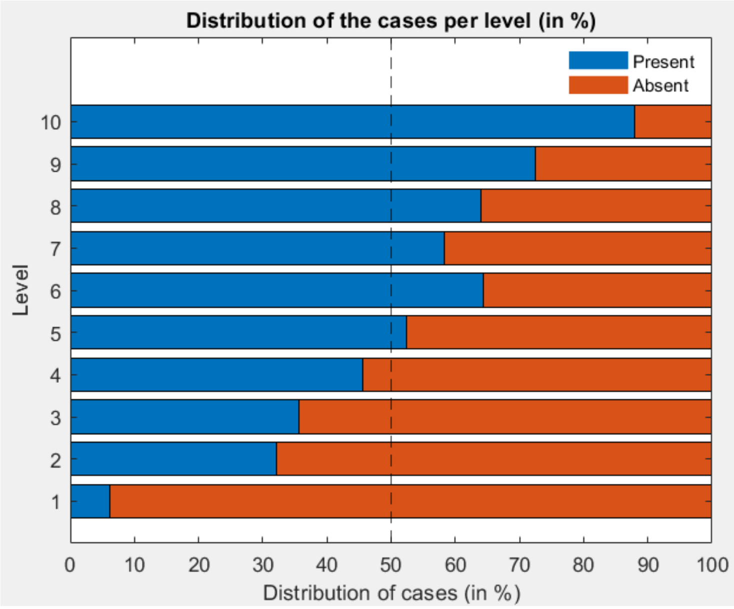


FIGURE 5.2: The distribution of the cases (present vs. absent) in each level, expressed in percentages.

5.1.4 Implementation heat maps

In the CAL-system occlusion sensitivity heat maps were used, which were generated by the top-performing DNN of the 70 DNNs. However, a better implementation would be to generate the heat map of three top-performing DNNs and merge these together, to eventually obtain a better approximation for the heat map. However, due to this requiring a lot of system resources and due to the long computation times, while also taking the time constraints of this research into account, this was unfortunately not performed.

It must be noted, however, that a significant amount of heat maps shown in the CAL-system may not be an accurate representation of the location of an ICH in an image. In this research, the top performing DNN had an accuracy of approximately 80%, meaning subsequently that one in five heat maps may show no information about the location of an ICH or is faulty in general. In the worst case scenario, this would give the user an incorrect point of reference when diagnosing an ICH case. Therefore, precautions should be taken when making use of heat maps, and preferably should only be used when the DNN used for the generation of heat maps has a significant high accuracy to do so.

Nonetheless, since other types of generating heat maps exist as described in *Theory & Background: Neural Networks*, in future research these other methods should also be looked at and should be evaluated by professional radiologists, to determine which of these methods generally provide the most accurate and most insightful heat maps.

5.1.5 Sample size and sampling method

In literature, 25 to 30 samples is considered the minimum sample size as a rule of thumb [30]. Unfortunately, this amount of participants was not achieved in this study: only 8 participants partook in the CAL-system. Therefore, the statistical power is significantly less than desired, consequently affecting the conclusions made about the effectiveness of the CAL-system.

Furthermore, the participants were selected based on convenience sampling. This method, however, results in samples that are generally not representatives of the target population, since this method is nonprobabilistic in nature.

All in all, in future research it is desired to test the CAL-system on more participants, to obtain results and respectively derive conclusions which are based on a stronger statistical power basis.

5.2 Minor aspects

5.2.1 Associated level of test cases

In the tests, the level associated with a case shown during the test had not been logged and was overlooked. This, however, would have provided valuable information about comparing the difficulty of cases associated with a certain level between the human response and that of the DNNs response. In future research, it is advised to log the associated level of a case in a test to gain a better understanding in how the response between humans and DNNs may differ.

5.2.2 Neglect of training on ICH types

One of the limits encountered early in this research is that in the CAL-system a user is not able to train their diagnostic skills based on the various types of ICH (these types are described in the Introduction), since the DNNs are not able to classify the type of ICH in an image, but only can classify if ICH is present in an image or not. In other words, the CAL-system can only be considered a general purpose training system for improving diagnostic skills of any type of ICH, and does not offer any training that specializes in the various ICH types. Therefore, it is recommended for future research that, if a DNN is able to classify the ICH type, this should be utilized to its fullest potential, since a CAL-system allowing the user to train their diagnostic skills based on specific ICH types may provide better opportunities to expand the user's diagnostic skills even more.

5.2.3 CT image quality in real world

A noteworthy aspect encountered early in this research was that the images themselves were of low quality, consisting of 128x128 pixel bitmap images. However, in the medical field CT images of higher quality are used, which are (uncompressed) 512x512 pixel bitmap images [28]. The low quality of the CT-images shown in the application may have influenced their diagnostic ability to correctly diagnose a case.

In other words, the experience of visually diagnosing an image in this CAL-system may differ when comparing it to the real world scenario. This in turn means that there may be a possibility that diagnostic skills obtained in this CAL-system may not fully translate into diagnostic skills in the real world. Therefore, in a future model of a CAL-system it is advised to make use of higher quality CT images, to better reflect the real world scenario.

5.2.4 Repurposing tests for CAL-system

In this research, the pre-training and post-training test were initially only included in the application to only evaluate the effectiveness of the CAL-training. However, the tests themselves can be repurposed and be incorporated in the CAL-system, functioning as auxiliary components.

The pre-training test can be considered a useful tool for the CAL-system to bring the user to a certain level based on their already existing experience when starting the CAL-training, which was already realized as part of the CAL-system in this research.

On the other hand, the post-training test can be repurposed in such a way that after a user reaches a level threshold, a test will be presented to the user. Here, the user will only be able to climb up the ranks again if they passed this test. By introducing this gatekeeping element, an extensive intermediate evaluation is performed which determines if the user's diagnostic skills actually improved or not.

5.2.5 Pedagogy and CAL-system

It is argued that it is the pedagogy that affects the outcome of the learning process and not the technology [29]. Furthermore, a CAL-system is considered more effective when only certain aspects of the CAL-system are addressed, such as the optimization of the feedback system and enabling the student to be in control of learning. In other words, the methods that are used for the CAL-system ultimately determines the effectiveness of its implementation, and not the medium that is used.

This would mean that poor performance of the CAL-system may eventually be attributed to respectively an implementation of poor methods. In this case, making a statement about the low or high potential of the developed CAL-system in this research becomes more uncertain, since it may be the case that a highly effective CAL-system can be realized if only if the correct pedagogy was implemented.

5.2.6 No feedback / suggestions of participants

A feedback and/or suggestion box should have been implemented at the end application, such that participants were able to give their feedback about their experience with the CAL-system or the application overall, to find out which aspects of the CAL-system and/or application positively or negatively contributed to their experience. It is of importance to ask this of the participants, since the user experience also may play an influential role in being able to efficiently operate the CAL-system, which in turn may affect the efficiency of the user obtaining diagnostic skills.

5.2.7 Conditions for effective CAL-system

In *Theory & Background: Computer-assisted learning (CAL)*, five conditions were mentioned that enables computer learning to be effective. One could argue that in this CAL-system the user is in control of the learning and that feedback is optimized as much as possible, since the user could solve and review cases with at their own pace while making sure that enough feedback is given to the user in the form of correct/incorrect feedback and heat maps, therefore fulfilling both conditions.

However, it could be argued that the other conditions were not fulfilled, since there was no peer learning present as the user solved cases on their own, nor was there diversity of teaching strategies or multiple opportunities for learning present in the CAL-system (as described in *Neglect of training on ICH types*). Therefore, in future research these unfulfilled conditions should be taken into consideration when further developing the CAL-system, such that the effectiveness of the CAL-system is optimized.

Chapter 6

Conclusion

The findings of this research suggest that the developed CAL-system has potential for improving ICH diagnostic skills in people with no medical background, as a significant increase in diagnostic performance was observed in this group, especially when it comes to their accuracy ($p = 0.0125$) and their sensitivity ($p = 0.04$), but not their specificity ($p > 0.05$). However, this positive trend was not observable in participants with a medical background, as no significant performance increase in the accuracy, sensitivity and/or specificity was observed in this group ($p > 0.05$).

It is unclear if the level system in the CAL-system is effective, as the findings of this research suggest no clear relation between a user reaching a higher (end) level with respect to their begin level and a user improving their diagnostic performance after CAL-training. Furthermore, it is unclear if item analysis, using deep neural networks as the "test-taking" group, is an effective method to gauge the difficulty of an ICH case for humans.

All in all, the findings suggest that the CAL-system developed in this research has potential for the training of freshmen radiology students, for which the CAL-system can be utilized as a quick crash course to bring their ICH diagnostic skills up to a certain baseline level.

Bibliography

[1] Gaillard, F. (2021). Intracranial hemorrhage — Radiology Reference Article — Radiopaedia.org. Retrieved 11 February 2021, from https://radiopaedia.org/articles/intracranial-haemorrhage

[2] Tenny, S., & Thorell, W. (2020). Intracranial Hemorrhage. In StatPearls. StatPearls Publishing.

[3] Strub, W. M., Leach, J. L., Tomsick, T., & Vagal, A. (2007). Overnight preliminary head CT interpretations provided by residents: locations of misidentified intracranial hemorrhage. AJNR. American journal of neuroradiology, 28(9), 1679–1682. doi:10.3174/ajnr.A0653

[4] Takagi, Y., Hadeishi, H., Mineharu, Y., Yoshida, K., Ogasawara, K., Ogawa, A., & Miyamoto, S. (2018). Initially Missed or Delayed Diagnosis of Subarachnoid Hemorrhage: A Nationwide Survey of Contributing Factors and Outcomes in Japan. Journal of stroke and cerebrovascular diseases : the official journal of National Stroke Association, 27(4), 871–877. doi:10.1016/j.jstrokecerebrovasdis.2017.10.024

[5] Arbabshirani, M. R., Fornwalt, B. K., Mongelluzzo, G. J., Suever, J. D., Geise, B. D., Patel, A. A., & Moore, G. J. (2018). Advanced machine learning in action: identification of intracranial hemorrhage on computed tomography scans of the head with clinical workflow integration. NPJ digital medicine, 1, 9. doi:10.1038/s41746-017-0015-z

[6] Morcos, S. K., & Suwais, M. (1996). Computer assisted learning for the interpretation of the chest radiograph. Health Informatics, 2(3), 146–148. doi:10.1177/146045829600200308

[7] Shaikh, F., Inayat, F., Awan, O., Santos, M. D., Choudhry, A. M., Waheed, A., Kajal, D., & Tuli, S. (2017). Computer-Assisted Learning Applications in Health Educational Informatics: A Review. Cureus, 9(8), e1559. doi:10.7759/cureus.1559

[8] Murphy, K. P. (2013). Machine learning : a probabilistic perspective. Cambridge, Mass. [u.a.]: MIT Press. ISBN: 9780262018029 0262018020

[9] Zeiler, M. D., & Fergus, R. (2014). Visualizing and Understanding Convolutional Networks. Computer Vision, 818-833. doi:10.1007/978-3-319-10590-1_53

[10] Understand Network Predictions Using Occlusion- MATLAB & Simulink. (2021). Retrieved 25 February 2021, from https://www.mathworks.com/help/deeplearning/ug/understand-network-predictions-using-occlusion.html

[11] Grad-CAM Reveals the Why Behind Deep Learning Decisions-MATLAB & Simulink. (2021). Retrieved 25 February 2021, from https://www.mathworks.com/help/deeplearning/ug/gradcam-explains-why.html

[12] Understand Network Predictions Using LIME- MATLAB & Simulink. (2021). 25 February 2021, from https://www.mathworks.com/help/deeplearning/ug/understand-network-predictions-using-lime.html

[13] Basic Concepts in Item and Test Analysis. (2021). Retrieved 19 February 2021, from http://ericae.net/ft/tamu/Espy.htm

[14] Bichi, Ado. (2015). Item Analysis using a Derived Science Achievement Test Data. International Journal of Science and Research (IJSR). 4. 1655-1662.

[15] Marie, S., & Edannur, S. (2015). Relevance of Item Analysis in Standardizing an Achievement Test in Teaching of Physical Science in B.Ed Syllabus. Journal on Educational Technology, 12, 30-36.

[16] Young, M., Cummings, B. A., & St-Onge, C. (2017). Ensuring the quality of multiple-choice exams administered to small cohorts: A cautionary tale. Perspectives on medical education, 6(1), 21–28. doi:10.1007/s40037-016-0322-0

[17] Paz, Marie & Morales, Ernesto. (2012). Development and Validation of a Concept Test in Introductory Physics for Biology Students. Manila Journal of Science. Volume 7. pp. 26-44.

[18] Hartati, Neti & Yogi, Hendro. (2019). Item Analysis for a Better Quality Test. English Language in Focus (ELIF). 2. 59. 10.24853/elif.2.1.59-70.

[19] Item Discrimination Indices. (2021). Retrieved 19 February 2021, from https://www.rasch.org/rmt/rmt163a.htm

[20] Badyal, D. K., Bala, S., Singh, T., Gulrez, G. (2019). Impact of immediate feedback on the learning of medical students in pharmacology. Journal of advances in medical education and professionalism, 7(1), 1–6. doi:10.30476/JAMP.2019.41036

[21] Fazio, L. K., Huelser, B. J., Johnson, A., Marsh, E. J. (2010). Receiving right/wrong feedback: consequences for learning. Memory (Hove, England), 18(3), 335–350. doi:10.1080/09658211003652491

[22] Butler, A., Godbole, N., Marsh, E. (2013). Explanation feedback is better than correct answer feedback for promoting transfer of learning. Journal Of Educational Psychology, 105(2), 290-298 doi:10.1037/a0031026

[23] Chamberland, M., Setrakian, J., St-Onge, C., Bergeron, L., Mamede, S., Schmidt, H. (2019). Does providing the correct diagnosis as feedback after self-explanation improve medical students diagnostic performance?. BMC Medical Education, 19(1). doi:10.1186/s12909-019-1638-3

[24] Watanabe, Y., Tanaka, T., Nishida, A., Takahashi, H., Fujiwara, M., Fujiwara, T. et al. (2020). Improvement of the diagnostic accuracy for intracranial haemorrhage using deep learning–based computer-assisted detection. Neuroradiology. doi:10.1007/s00234-020-02566-x

[25] Sandercock, G. (2009). Computer assisted learning as an effective way of teaching and learning in medical education.

[26] De Bruyckere P., Kirschner P.A. (2020) Computer-Assisted Learning. In: Tatnall A. (eds) Encyclopedia of Education and Information Technologies. Springer, Cham. doi:10.1007/978-3-030-10576-1_73

[27] Martínez-Mesa, J., González-Chica, D. A., Duquia, R. P., Bonamigo, R. R., & Bastos, J. L. (2016). Sampling: how to select participants in my research study?. Anais brasileiros de dermatologia, 91(3), 326–330. doi:10.1590/abd1806-4841.20165254

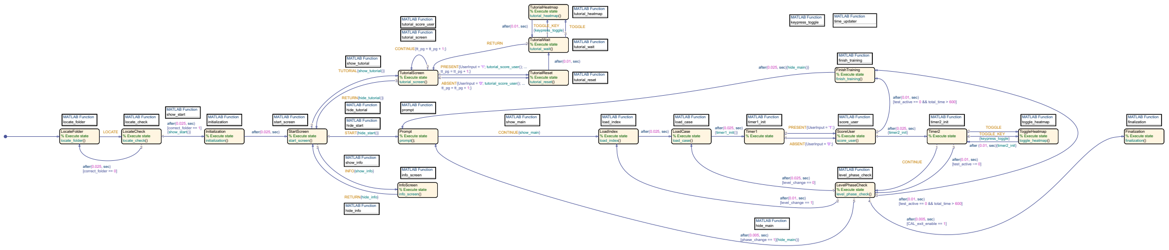
[28] Liu, Feng & Hernández-Cabronero, Miguel & Sanchez, Victor & Marcellin, Michael & Bilgin, Ali. (2017). The Current Role of Image Compression Standards in Medical Imaging. Information. 8. 131. 10.3390/info8040131.

[29] De Bruyckere P., Kirschner P.A. (2020) Computer-Assisted Learning. In: Tatnall A. (eds) Encyclopedia of Education and Information Technologies. Springer, Cham. doi:10.1007/978-3-030-10576-1_73

[30] Hogg, R., Tanis, E., Zimmerman, D. (2015). Probability and statistical inference (Ninth edition.). Pearson.

Appendix

Appendix A: MATLAB Stateflow state machine of the developed application


MATLAB Function locate_folder
LocateFolder % Execute state locate_folder()
LOCATE
MATLAB Function locate_check
LocateCheck % Execute state locate_check()
after(0.025, sec) [correct_folder == 0]
MATLAB Function show_start
after(0.025, sec) [correct_folder == 1] {show_start()}
MATLAB Function initialization
Initialization % Execute state initialization()
after(0.025, sec)
MATLAB Function start_screen
StartScreen % Execute state start_screen()
MATLAB Function show_tutorial
TUTORIAL{show_tutorial()}
RETURN{hide_tutorial()}
MATLAB Function hide_tutorial
MATLAB Function hide_start
START{hide_start()}
MATLAB Function show_info
INFO{show_info}
RETURN{hide_info}
MATLAB Function hide_info
MATLAB Function info_screen
InfoScreen % Execute state info_screen()
MATLAB Function tutorial_score_user
MATLAB Function tutorial_screen
CONTINUE{tt_pg = tt_pg + 1;}
TutorialScreen % Execute state tutorial_screen()
PRESENT{UserInput = '1'; tutorial_score_user(); ... tt_pg = tt_pg + 1;}
ABSENT{UserInput = '0'; tutorial_score_user(); ... tt_pg = tt_pg + 1;}
RETURN
TutorialHeatmap % Execute state tutorial_heatmap()
MATLAB Function tutorial_heatmap
after(0.01, sec)
TOGGLE_KEY {keypress_toggle}
TOGGLE
TutorialWait % Execute state tutorial_wait()
MATLAB Function tutorial_wait
after(0.01, sec)
TutorialReset % Execute state tutorial_reset()
MATLAB Function tutorial_reset
MATLAB Function prompt
Prompt % Execute state prompt();
MATLAB Function show_main
CONTINUE{show_main}
MATLAB Function load_index
LoadIndex % Execute state load_index()
after(0.025, sec)
MATLAB Function load_case
LoadCase % Execute state load_case()
after(0.025, sec) {timer1_init()}
MATLAB Function timer1_init
Timer1 % Execute state
PRESENT{UserInput = '1';}
ABSENT{UserInput = '0';}
MATLAB Function score_user
ScoreUser % Execute state score_user()
after(0.025, sec) {timer2_init}
MATLAB Function timer2_init
Timer2 % Execute state
TOGGLE
TOGGLE_KEY {keypress_toggle}
after (0.01, sec){timer2_init}
MATLAB Function toggle_heatmap
ToggleHeatmap % Execute state toggle_heatmap()
MATLAB Function keypress_toggle
MATLAB Function time_updater
MATLAB Function finish_training
after(0.025, sec){hide_main()}
FinishTraining % Execute state finish_training()
after(0.01, sec) [test_active == 0 && total_time > 600]
CONTINUE
after(0.01, sec) [test_active ~= 0]
MATLAB Function level_phase_check
after(0.025, sec) [level_change == 0]
after(0.01, sec) [level_change == 1]
LevelPhaseCheck % Execute state level_phase_check()
after(0.01, sec) [test_active == 0 && total_time > 600]
after(0.005, sec) [CAL_exit_enable == 1]
MATLAB Function hide_main
after(0.005, sec) [phase_change == 1]{hide_main()}
MATLAB Function finalization
Finalization % Execute state finalization()