UPPSALA
UNIVERSITET

UPTEC STS 21040
Examensarbete 30 hp
December 2020

Automation Pipelines for Efficient and Robust Experimental Research Within Cognitive Neuroscience

Patrik Björklund
Anna Rydin

Civilingenjörsprogrammet i system i teknik och samhälle

UPPSALA UNIVERSITET

Automation Pipelines for Efficient and Robust Experimental Research Within Cognitive Neuroscience

Patrik Björklund
Anna Rydin

Abstract

The current trend towards large-scale research projects with big quantities of data from multiple sources require robust and efficient data handling. This thesis explores techniques for automatizing research data pipelines. Specifically, two tasks related to automation within a long-term research project in cognitive neuroscience are addressed. The first task is to develop a tool for automatic transcribing of paper-based questionnaires using computer vision. Questionnaires containing continuous scales, so called visual analog scales (VASs), are used extensively in e.g. psychology. Despite this, there currently exists no tool for automatic decoding of these types of questionnaires. The resulting computer vision system for automatic questionnaire transcribing we present, called "VASReader", reliably detects VAS marks with an accuracy of 98%, and predicts their position with a mean absolute error of 0.3 mm when compared to manual measurements. The second task addressed in this thesis project is to investigate whether machine learning can be used to detect anomalies in Magnetic Resonance Imaging (MRI) data. An implementation of the unsupervised anomaly detection technique Isolation Forest shows promising results for the detection of anomalous data points. The model is trained on image quality metric (IQM) data extracted from MRI. However, it is concluded that the site of scanning and MRI machine model used affect the IQMs, and that the model is more prone to classify data points originating from machines and institutions that have less support in the database as anomalous. An important conclusion from both tasks is that automation is possible and can be a great asset to researchers, if an appropriate level and type of automation is selected.

Teknisk-naturvetenskapliga fakulteten

Uppsala universitet, Utgivningsort Uppsala/Visby

Handledare: William Thompson Ämnesgranskare: Anders Brun

Examinator: Elísabet Andrésdóttir

Populärvetenskaplig sammanfattning

Det här arbetet har gjorts i samarbete med en forskargrupp inom fältet kognitiv neurovetenskap på Karolinska Institutet. Med hjälp av olika tekniker för hjärnavbildning fokuserar forskargruppen på smärta och dess underliggande mekanismer. Ett av deras forskningsprojekt, kallat PrePain, bygger på storskalig och långvarig datainsamling. Genom att samla in data från funktionell magnetresonanstomografi (fMRT), enkäter, och nationella register, ämnar forskarna identifiera riskfaktorer som kan leda till utveckling av långvarig smärta. Studien är prospektiv, och stora mängder data från friska individer kommer samlas in under flera år för att se samband mellan riskfaktorer och framtida utveckling av kronisk smärta. Den långvariga och storskaliga datainsamlingen, samt det faktum att viss fMRT-data från externa forskargrupper kommer utnyttjas, ställer krav på effektiv datahantering. Det är inte bara inom projektet PrePain som effektiv och robust databehandling är angeläget, utan en generell utveckling ses inom vetenskapen mot allt mer storskaliga forskningsprojekt med data från multipla källor.

För att effektivisera databehandlingen vill forskarna bakom PrePain bygga en pipeline, det vill säga skapa en seriestruktur av databehandlingselement. Målet är att automatisera tidskrävande och felbenägna datahanteringsmoment i pipelinen.

Särskilt två uppgifter inom forskningsdatapipelinen behandlas i den här uppsatsen. För det första undersöks möjligheten att automatiskt avkoda de enkäter som fylls i av studiedeltagare. Enkäterna är pappersbaserade och innehåller så kallade visuella analoga skalor, förkortat VAS. En visuell analog skala representeras av en 100 millimeter lång linje, som används för självuppskattning av bland annat smärta. Studiedeltagare markerar sitt svar genom att sätta ett kryss eller streck på linjen. VAS är ett validerat psykometriskt mått, och trots att det är vanligt förekommande inom smärtforskning och psykologi, saknas verktyg för att automatiskt läsa av VAS-enkäter. I nuläget mäts varje enskild markerings position för hand med en linjal, och resultatet förs därefter in i en databas. Det nuvarande tillvägagångssättet är mycket tidskrävande och även felbenäget eftersom stor noggrannhet krävs. Med anledning av detta har ett delmoment inom det här projektet varit att utveckla tekniker för att automatiskt läsa av inskannade enkäter. Att extrahera information från digitala bilder ingår i forskningsområdet *datorseende*, på engelska *computer vision.*

Den andra uppgiften inom projektets ramar är istället kopplad till en annan datatyp, nämligen magnetresonanstomografibilder (MR-bilder). Funktionell MR (fMRT) används för att kartlägga smärtsignaler i hjärnan. Brus och artefakter, det vill säga bildförvrängningar, är vanligt förekommande i MR-bilder. De kan till exempel orsakas av att studiedeltagaren rör sig under MR-skanningen. Det är viktigt att identifiera MR-bilder av undermålig kvalitet, för att exkludera dessa från fortsatt analys. Ett existerande verktyg, MRIQC, extraherar kvalitetsmått från MR-data. Kvalitetsmåtten är dock så pass

många att det är svårt att få en överblick över dem och göra en sammanvägd kvalitetsbedömning. Stora mängder data med kvalitetsmått finns att tillgå i en publik databas, vilket i det här projektet har utnyttjats för att träna en maskininlärningsmodell att detektera avvikande datapunkter. Maskininlärningsalgoritmen som har använts är en oövervakad teknik som heter *isolation forest.* Grundidén bakom isolation forest är att om anomalier antas vara få och skiljer sig från majoriteten av övriga data, är de också enklare att isolera.

Resultaten från projektet visar att det är möjligt med automatisk avläsning av enkäter, samt att avvikelsedetektering i kvalitetsmåttsdata från MR-bilder kan användas som en indikation på MR-datas kvalitet. Det system som i och med projektet har utvecklats för enkätavläsning har fått namnet *VASReader*. Med tanke på avsaknaden av liknande verktyg som läser av just VAS-enkäter, är förhoppningen att systemet kan användas även utanför PrePain-projektets ramar. *VASReader* har potential att bespara forskare mycket tidskrävande och monotont arbete, och frigör tid för dem att istället ägna sig åt sina forskningsämnen. *VASReader* innebär också att VAS-resultat registreras på ett systematiskt och exakt vis.

Att utvärdera resultat från oövervakade maskininlärningsmodeller är svårt, eftersom inget rätt svar finns. Visualiseringar av isolation forest-resultat visar dock att synligt avvikande datapunkter klassificeras som anomalier, samt att det mått på avvikelse som fås ut av algoritmen speglar de underliggande kvalitetsmåtten. Modellen visar dock tendenser till att klassa datapunkter från vanligt förekommande MR-skannermodeller och institutioner som mer normala än datapunkter från MR-skannermodeller och institutioner som är mindre frekvent förekommande i databasen. Mer forskning krävs också för att undersöka sambandet mellan avvikande datapunkter och faktiskt bildkvalitet.

Acknowledgement

This thesis project has been conducted in collaboration with the Pain Neuroimaging Lab at Karolinska Institutet. We would like to express our very great appreciation to our supervisor William Thompson, who has given us valuable help, support and encouragement throughout the project. Our special thanks are extended to all researchers at Pain Lab who have welcomed us into their research group and introduced us to cognitive neuroscience research. Finally, we wish to acknowledge the assistance provided by Anders Brun, our subject reader at Uppsala University.

Patrik Björklund & Anna Rydin

Uppsala, December 2020

Abbreviations

BOLD	Blood Oxygenation Level Dependent
CNN	Convolutional Neural Network
CV	Computer Vision
EDA	Exploratory Data Analysis
fMRI	functional MRI
ICR	Intelligent Character Recognition
IQM	Image Quality Metrics
KI	Karolinska Institutet
MNIST	Modified National Institute of Standards and Technology
MRI	Magnetic Resonance Imaging
MRIQC	Magnetic Resonance Imaging Quality Control tool
OCR	Optical Character Recognition
OMR	Optical Mark Recognition
OpenCV	Open source Computer Vision library
SOMR	Software Optical Mark Recognition (system)
T1	T1-weighted
t-SNE	t-distributed Stochastic Neighbor Embedding
VAS	Visual Analog Scale

Table of Contents

1. Introduction

PrePain is a unique large-scale database project within neuroimaging at Karolinska Institutet, aiming to find baseline factors that predict development of chronic pain later in life. The goal of PrePain is to predict who is at risk of developing long-term pain with a combination of brain images, genetic data, registry data and questionnaire data. In the coming years, thousands of healthy volunteers will have their brains scanned to collect structural and functional brain images using magnetic resonance imaging (MRI). In connection with this, participants fill in questionnaires containing continuous scales, so called visual analog scales (VAS), as well as binary YES/NO questions. Large-scale and long-term data collection projects such as PrePain require efficient strategies to collect and process the data as multiple different researchers will be involved through the project.

Due to the large-scale nature of the project, the researchers at the Pain Neuroimaging Lab at Karolinska Institutet wish to streamline the data management and build an automation pipeline in order to enhance efficiency and robustness. The pipeline should include elements such as collection of raw data from multiple sources, data conversion and standardization, preprocessing of MRI data, quality control, evaluation, error flagging and alerts when manual checks are needed. An outline of the pipeline can be seen below in Figure 1.

The main focus of this thesis project is to develop a tool for automatic reading and recording of questionnaires. Today, the researchers need to measure and transfer results from questionnaires manually with a ruler, a task that is time-consuming, error-prone and requires exactitude. An automatic solution can improve robustness, efficiency, and enable the researchers to spend more time on their subjects of interest. Despite their extensive usage within the research community, there exists no tool for automatic decoding of paper-based VAS questionnaires. A more extensively studied area that is related to the task of decoding VAS questionnaires is however the task of decoding marks from optical mark recognition (OMR) sheets. There is also done a significant amount of relevant similar work within the areas of image document processing, intelligent character recognition, and computer vision.

In addition to the need of a document-to-dataframe routine for questionnaire data, the researchers wish to investigate methods for automatic quality control of functional Magnetic Resonance Images (fMRI). The researchers behind PrePain have the opportunity to acquire data from other research groups. The large amounts of imaging data that will be acquired, and the fact that parts of it comes from external sources, increase the need for automatic identification of subpar images. These may, if they are not identified, cause problems in downstream analysis. There already exists a tool for extracting quality metrics from fMRI data, called MRIQC, developed by Esteban *et al.* (2017). However, MRIQC returns more than 50 image quality metrics (IQMs), which is a sufficiently large number for it to be difficult for a human to get an overview and make

an overall assessment of the quality. MRIQC reports do give an overview of how the quality metrics in a group of images relate to the other images' IQMs in that same group. Nevertheless, a tool that enables comparison of the retrieved IQMs with data from other scanning sites is thought to give a better indication of the quality. If substandard images are assumed to be few and different from most of the data, and this is reflected in the IQMs, an approach to identify and flag these images is by applying anomaly detection techniques using machine learning. Anomaly detection, or outlier detection, is the task of identifying observations that differ significantly from other data points, and examples of previous applications are fraud detection and intrusion detection.

The overarching aim of this project is to investigate whether time-consuming and error-prone manual tasks within cognitive neuroscience research can be automated by a pipeline. More specifically, the following two research questions will be addressed:

- Can questionnaires containing visual analog scales be read and decoded using computer vision?
- Can MRIQC data with Image Quality Metrics be utilized to detect anomalies in fMRI images?

The thesis project is hence divided into two separate subtasks, the first one within the field of computer vision, the second one within machine learning. The two subtasks' methodologies, data, and results are thus presented separately in Section 3 and 4. Albeit the two tasks require differing methodologies, they are both parts within the more general task of automatizing research pipelines and analyzing different types of images (scanned documents and MRI images). An outline of the planned pipeline can be seen in Figure 1. The tasks related to the first research question is highlighted in yellow, and the task related to the second question is highlighted in blue. The project has been conducted in collaboration with the Pain Neuroimaging Lab at Karolinska Institutet (KI). Regular contact and updating about the progression of the project has taken place through weekly meetings with the multidisciplinary research group and supervisor at KI. Halfway through the project, a presentation was held for the researchers in the research group to communicate the purpose of the work and show preliminary results. The methods used to solve the task were also presented. Visits on site at KI provided an opportunity to observe and have conversations with the researchers about how they collect data.

The project is delimited in that the questionnaire decoding tool is developed to work on the specific questionnaires used in the PrePain study. The resulting system is consequently not expected to work out of the box on all variations of questionnaires containing VASs. However, the intention is for the tool to be applicable to other questionnaires with some minor adjustments. The MRI quality control part of the project is delimited in that it is not an extensive comparison of different anomaly detection techniques, but rather an explorative data analysis (EDA) followed by evaluation and analysis of one selected machine learning model for outlier detection, called Isolation Forest, applied to both structural and functional MRI data.

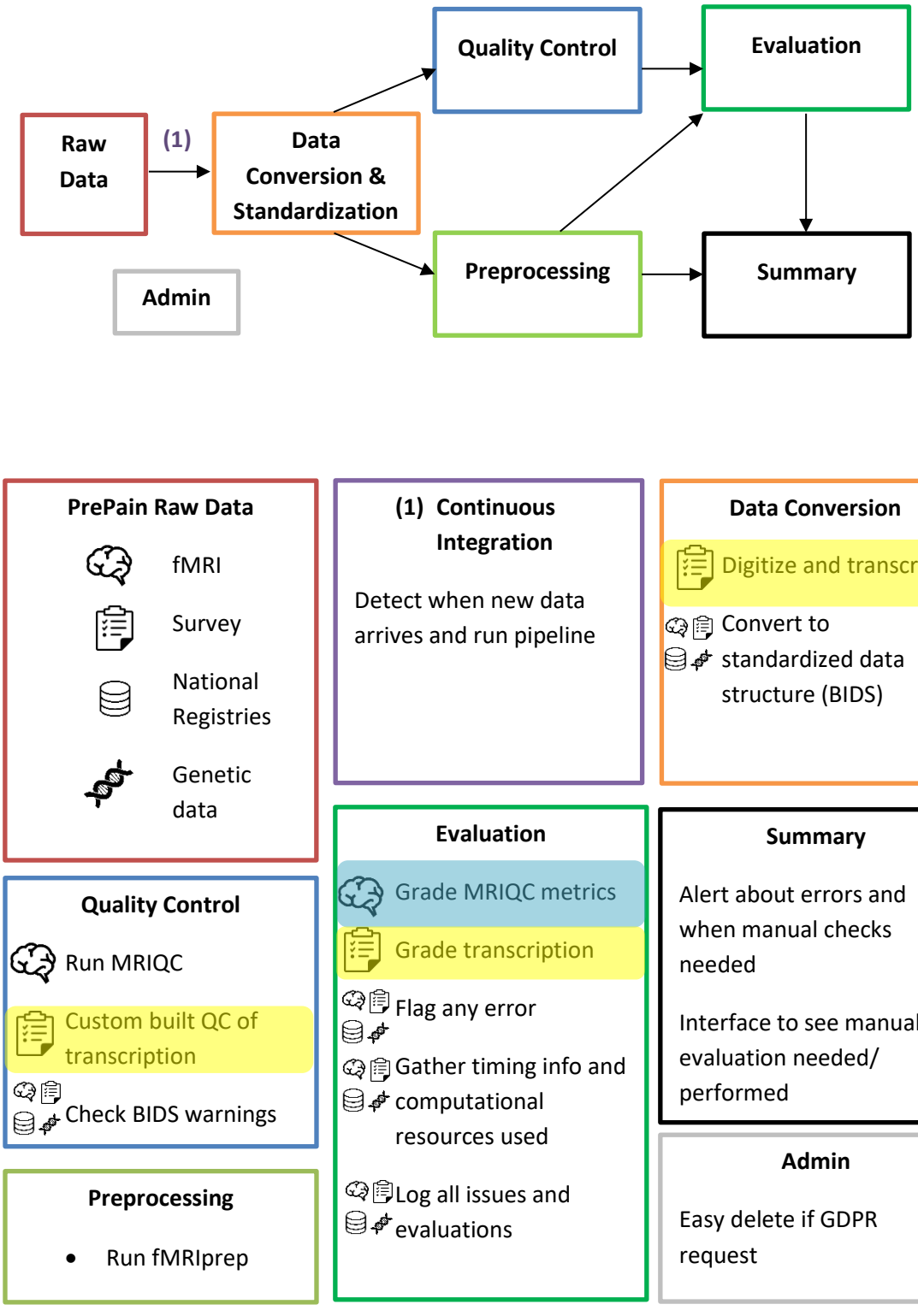


Figure 1. An outline of the PrePain pipeline. The tasks related to this project are highlighted in yellow and blue.

2. Background

This project involves both automation, specific components of image processing, computer vision and machine learning. For this reason, this background covers broad theories about system automation, as well as technical descriptions of computer vision algorithms and machine learning models. The data the automation tasks applies to – questionnaires and MRIQC image quality metrics, is also introduced.

2.1 Appropriate Selection of Type and Level of System Automation

The overarching aim of this project considers automatizing research pipelines. Thus, it is important to understand what automation means, the different types of automation that are possible, and how automation can be implemented and evaluated. Parasuraman et al. define automation as "a device or system that accomplishes (partially or fully) a function that was previously, or conceivably could be, carried out (partially or fully) by a human operator" (Parasuraman, Sheridan and Wickens, 2000). This definition entails that automation is not all-or-none, but a continuum of levels between no automation and full automation. Technical development in both hardware and software lead to the possibility to automate many previously human-performed tasks and operations in systems. In fact, there are few functions that cannot be automated (ibid).

Automation not only replaces human activity, but changes it, and Parasuraman et al. argue for the importance of asking questions such as "which system functions should be automated?" and "to what extent?". They present a framework for categorizing classes of functions and levels of automation, which is meant to aid appropriate selection of type and level of system automation (ibid).

Parasuraman et al. suggest that automation can be applied to four broad categories of functions in a human-machine system:

1) information acquisition
2) information analysis
3) decision selection
4) action implementation

These correspond to a simple conceptual four-stage model of human information processing, namely: sensory processing – perception/working memory – decision making – response selection (ibid).

Sheridan et al. (1978) proposed a 10-point scale which describes different levels of automation for decision selection and action implementation, see Table 1. This scale helps illustrate the continuous nature of automation defined above.

Parasuraman et al. (2000) adopt and extend the scale by arguing that it is also applicable to the input functions; information acquisition and information analysis. Thus, the four classes of system functions, 1) – 4) above, can be automated to different degrees, each ranging from fully manual performance to full automation.

Table 1. Levels of Automation of Decision and Action Selection (Sheridan, Verplank and Brooks, 1978; Parasuraman, Sheridan and Wickens, 2000)

HIGH	10.	The computer decides everything, acts autonomously, ignoring the human.
	9.	informs the human only if it, the computer, decides to
	8.	informs the human only if asked, or
	7.	executes automatically, then necessarily informs the human, and
	6.	allows the human a restricted time to veto before automatic execution, or
	5.	executes that suggestion if the human approves, or
	4.	suggests one alternative
	3.	narrows the selection down to a few, or
	2.	The computer offers a complete set of decision/action alternatives, or
LOW	1.	The computer offers no assistance: human must take all decisions and actions.

Along with which areas that can be automatized and the degree of the automation, there are also different criteria to evaluate the effectiveness of the automation. Parasuraman et al (2000) propose a model for automation design that is not based on technological capabilities or economic incentives. Instead, it considers two other evaluative criteria. The first evaluative criteria are the *human performance consequences*, and the second are *automation reliability and costs of decision and action consequences* (ibid).

The primary evaluation criteria used to evaluate a certain level of automation is about assessing the associated human performance consequences of the resulting system. Four areas of human performance are considered; mental workload, situation awareness, complacency, and skill degradation. Well-designed automation systems should decrease *mental workload*, not increase it. Ways in which automation can expand rather than eliminate problems for the human operator are discussed by Bainbridge (1983) Implementing so called "clumsy" automation, which essentially does more harm than good, tends to happen when automation is difficult to initiate and engage in, thus leading to both physical and mental additional work for the human system operator. *Situation awareness* refers to the operators' ability to get an overview of the system states and its information sources. If decision making is automated, it can be difficult for the operator to get a good picture of the system, as he or she is not actively participating in the process of evaluating information sources leading to a decision; hence the process becomes more "black boxed". At the same time, appropriate and well-designed automation of

information acquisition and analysis may facilitate the human's situation awareness. *Complacency*, or "over-trust" can occur when automation is high, but not perfectly reliable. Occasional automation fails can then go under the radar. *Skill degradation* can occur when a function is consistently handled automatically, and the human operator starts forgetting how to perform the function, i.e. their cognitive skills degrade. Skill degradation, as well as complacency and reduced situation awareness, mainly pose a threat in the event of a system failure that requires the human to act. They can cause the operator to show "out-of-the-loop" unfamiliarity with highly automated systems (Parasuraman, Sheridan and Wickens, 2000).

The secondary evaluative criteria proposed by Parasuraman et al (2000) are automation reliability and costs of decision/action outcomes. An automated system's *reliability* highly affects the human trust of it and subsequent use of it. If high reliability cannot be ensured, the benefits on mental workload and situation awareness described earlier are not likely to hold. There are several methods to estimate automation reliability, such as software reliability analysis and fault testing. However, complexity and size of software, unpredictable failures, and faults arising from interaction with other existing systems impose difficulties in estimating and ensuring automation reliability. Mistrust in automated systems may lead to underutilization or disabling of the automated system. Only if information automation is highly reliable, high levels of automation are justified. (Parasuraman, Sheridan and Wickens, 2000). According to Lee and See (2004), people respond to technology socially, and because of this, trust influences reliance on automation. This is especially the case when complexity and unanticipated situations make a complete understanding of the automation impractical.

Evaluating *costs of decision/action outcomes* is about weighting the risk associated with incorrect or inappropriate decisions or actions against the benefits of a particular level of automation. Parasuraman et al. here adopts a definition of risk as the probability of an error multiplied by the cost of that error. Decisions involving small risks are candidates for high levels of automation in decision selection and action implementation, while highly automated high-risk decisions and actions are only motivated in, for example, extremely time-critical situations (ibid). In the case of highly automated decision selection, it is suggested giving the human the "last saying ", i.e. forcing the user to consciously confirm the decision before the action is implemented (ibid). Data entry is an example of a viable candidate for high automation in decision selection, with some degree of human involvement for error trapping (ibid). Studies on flight management systems – FMSs – in aircrafts have shown that a lower level of action automation in entering data into a computer (manual entry) had a higher error rate than a higher level of automation when the operator confirmed automated decisions by pressing an "accept" button to enter data (Hahn and Hansman, 2013).

2.2 Using Computer Vision to Extract Information from Images

Computer vision (CV) is a scientific field of study engaged in questions concerning how computers can “see” and “understand” digital images (and video). The aim of computer vision is to extract useful information from images in an automated way. What useful information is varies from application to application, and CV has applications in various fields such as medical imaging, surveillance, autonomous vehicles, and optical character recognition (OCR). Computer vision is a multidisciplinary field, with relations to e.g. artificial intelligence, neurobiology, information engineering, and signal processing. It is also closely related to image processing, image analysis and machine vision, and a combination of techniques originating from these fields are used in computer vision. A commonly used library of programming functions for computer vision is *OpenCV*, short for *Open Source Computer Vision Library*.

The task of automatically decoding questionnaires requires techniques from the fields of computer vision and image document processing. Even though the specific case of reading marks on VASs has not been explicitly addressed previously, it can to a great extent be solved using existing CV- and document image processing techniques. Especially relevant are methods for image preprocessing, skew correction, line detection, mark detection, shape detection etc. Previous research in image document processing in general, and OMR, OCR and ICR in particular, tackle many problems also encountered in solving the computer vision task of this thesis project. For this reason, an introduction to the fields of study that are relevant to this project will be given in Section 2.2.3, along with examples of previous relevant research. First, however, the PrePain questionnaires will be described shortly. Following this, the very basics of digital images and common operations on them are introduced. An understanding of how images are represented in computers is required to understand the techniques described in the coming sections.

2.2.1 PrePain Questionnaires

The main questionnaire used in the PrePain project consists of five binary answer questions and seventeen VAS questions. The questionnaire has two pages. The design of the questionnaires changed during the project, but the contents are identical. The changes, and the motivation behind them will be further described in Section 3. The initial questionnaire design is hereafter referred to as the first version, and a full version can be found in Appendix A. The altered questionnaire is hereafter referred to as the second version. It is found in Appendix B.

The binary questions consist of a description, a question, and the two answer alternatives as seen in Figure 2 below. JA and NEJ (Eng. YES and NO) are the answer alternatives. In the first version no instruction for how to mark an answer is provided.

Långvarig smärta

Långvarig smärta är den typ av smärta som inte går att förklara med tillfällig huvudvärk, träningsvärk, mensvärk eller liknande. Långvarig smärta varar i mer än tre månader i sträck.

- Lider du just nu av långvarig smärta? JA NEJ
- Har du tidigare haft perioder med mycket smärta i kroppen? JA NEJ
- Har du någon i din nära familj som lider av långvarig smärta? JA NEJ
- Är du orolig för att drabbas av långvarig smärta? JA NEJ

Figure 2. Four of the five binary questions with a common description as they were stated in the first version of the main questionnaire.

The VAS questions are divided into two sections. The first section consists of an overarching question; "Are you feeling pain anywhere in your body right now?", and eleven VASs of pain intensity for different areas of the body. The second section consists of six individual questions with one corresponding VAS of some measure each. An example from the second section can be seen in Figure 3 below.

- Hur jobbigt tycker du att det är när du har ont?

[----------]

Väldigt jobbigt Inte alls jobbigt

Figure 3. One of the six questions from the second VAS section as it was stated in the first version of the main questionnaire.

2.2.2 Computer Vision Foundations

Scanned document images can be decoded using computer vision. In computers, images are represented as matrices of pixel values. Images can thus be defined as a function f, mapping from R^2 to R. In the case of grayscale images, each pixel can typically take an intensity value in the range [0, 255], where 0 is black and 255 white. The pixel value $f(x, y)$ gives the intensity of the image at position (x, y). For colored images, the additive color model RGB, short for red, green and blue, is often used. For RGB images, $f(x, y)$ is a vector of three values, each in range [0, 255], see formula 1.

$$f(x, y) = \begin{bmatrix} r(x, y) \\ g(x, y) \\ b(x, y) \end{bmatrix} \quad (1)$$

A simple, yet effective method for segmenting images is thresholding. In a grayscale image, thresholding can be used to create a binary image, partitioned into background and foreground. There are various thresholding techniques. In simple thresholding, each pixel

in the image is replaced with a black pixel if the intensity of that pixel is less than a fixed threshold $thres$, or else it is replaced with a white pixel. That is,

$$dst(x,y) = \begin{cases} maxVal, & if\ src(x,y) > thres \\ 0, & otherwise \end{cases}, \quad (2)$$

where $src(x,y)$ is the intensity of the pixel at (x,y), and $dst(x,y)$ is the new intensity of pixel (x,y) (*OpenCV: Basic Thresholding Operations*, no date). The maximum pixel intensity $maxVal$ is typically 255.

The result of this operation is a binary image, consisting of only black and white pixels. An inverted binary thresholding operation (ibid.) can be expressed as

$$dst(x,y) = \begin{cases} 0, & if\ src(x,y) > thres \\ maxVal, & otherwise \end{cases}. \quad (3)$$

Inverted binarization is useful if one wishes to retrieve images in a binary format where black pixels have intensity 255, and white pixels intensity 0. Another thresholding technique is adaptive thresholding, which works better than simple thresholding for images with varying illumination. In adaptive thresholding, as opposed to using a global constant, the threshold for a pixel is based on the intensity values in the local neighborhood of that pixel (*OpenCV: Image Thresholding*, no date).

Doing operations on images represented as matrices of pixels with intensity values form the very basis of how computers can gain a high-level understanding of images. State-of-the-art techniques include usage of convolutional neural networks, machine learning models that convolve over images and extract features from them. Irrespective of the complexity of the computer vision method used for a task, they all build upon the data representation of digital images presented above. Neural network methods are not always the preferred option, inter alia due to their need of large quantities of training data. Often the complex methods are combined with basic image operations for preprocessing of images. Common problems addressed in computer vision are different types of detection, recognition and identification. Optical mark recognition (OMR), optical character recognition (OCR) and intelligent character recognition (ICR) are three fields within the broader field of computer vision. To give an overview of the context in which the questionnaire decoding task of this project belongs to, a description of these fields, and previous work within them follows in the next section.

2.2.3 Previous Work in OMR, OCR, and ICR

Optical mark recognition or optical mark reading – often referred to as OMR – is the process of decoding human-made marks from document forms like surveys, tests or

election ballots. OMR is for example used to read data from multiple choice question exams and questionnaires where answers are represented by crosses in boxes or filled-in bubbles (see e.g. Barney Smith, Nagy and Lopresti, 2009; Afifi and Hussain, 2017; Loke, Kasmiran and Haron, 2018).

The traditional OMR systems, which are hardware-based and require designated OMR scanners, have increasingly been replaced by software optical mark recognition systems – SOMR systems (Loke, Kasmiran and Haron, 2018). Benefits in flexibility, time efficiency, and costs have been given as reasons behind this development (Fisteus, Pardo and García, 2013). Unless otherwise stated, OMR techniques will hereafter mainly refer to software-based systems, as the methods used by traditional OMR scanners differ considerably from software-based OMR techniques.

A distinction is made between OMR and optical character recognition (OCR). In general, OMR is used to recognize darkened marks at predefined positions (Loke, Kasmiran and Haron, 2018), while an OCR system analyzes the shape of marks in an image and the recognizes the characters (Butterfield, Ngondi and Kerr, 2016). Simply, OMR detects the presence of a mark, while OCR decodes the shape of the mark. A more advanced form of OCR is called ICR, intelligent character recognition. Machine learning models based on neural networks, such as convolutional neural networks (CNNs), long short-term memory networks (LSTMs) and recurrent neural networks (RNNs) are often used in ICR (Ptucha *et al.*, 2019). Even though OCR formally includes deciphering of human-written characters, OCR today often refers to recognition of machine generated text while ICR refers to recognition of handwritten symbols (ibid).

Previous computer vision research that is relevant for this project include Fisteus, Pardo and García's paper introducing *Eyegrade*, a system for automatic grading of multiple choice exams (2013). The system is composed by six sequential steps:

1) Preprocessing by transforming the image to monochrome, and using adaptive thresholding;
2) Line detection with Hough transform;
3) Answer table detection by finding intersecting lines;
4) Decision making;
5) Exam model detection; and
6) Student ID detection.

Another example is Yan Ping Zhou and Chew Lim Tan' s (2000) usage of modified probabilistic Hough line transform to detect and recognize bar charts in document images. De Lima *et al.*, (2016) present an algorithm that finds signature lines in documents. Their system begins with preprocessing using simple thresholding. This is followed by candidate line detection using Hough lines and "connected components" to merge detected lines belonging to the same line. The last step is classification using Histogram of Gradient (HoG) features and Euclidian distance measure (ibid.). Elias, Tasinaffo and Hirata (2019) propose a method to handle skew, translation, scale and

alignment using a reference document in which key points are found by a pattern matching algorithm. The intention is for the method to be used before OMR.

A common feature in previous research is that the systems follow a similar workflow. With some deviations, the systems tend to be built as sequential processes containing the following main components: 1. Image document preprocessing; 2. Element detection, and 3. Classification, see Figure 4. Another common feature is the usage of a reference document to handle skew, alignment and translation of the images.

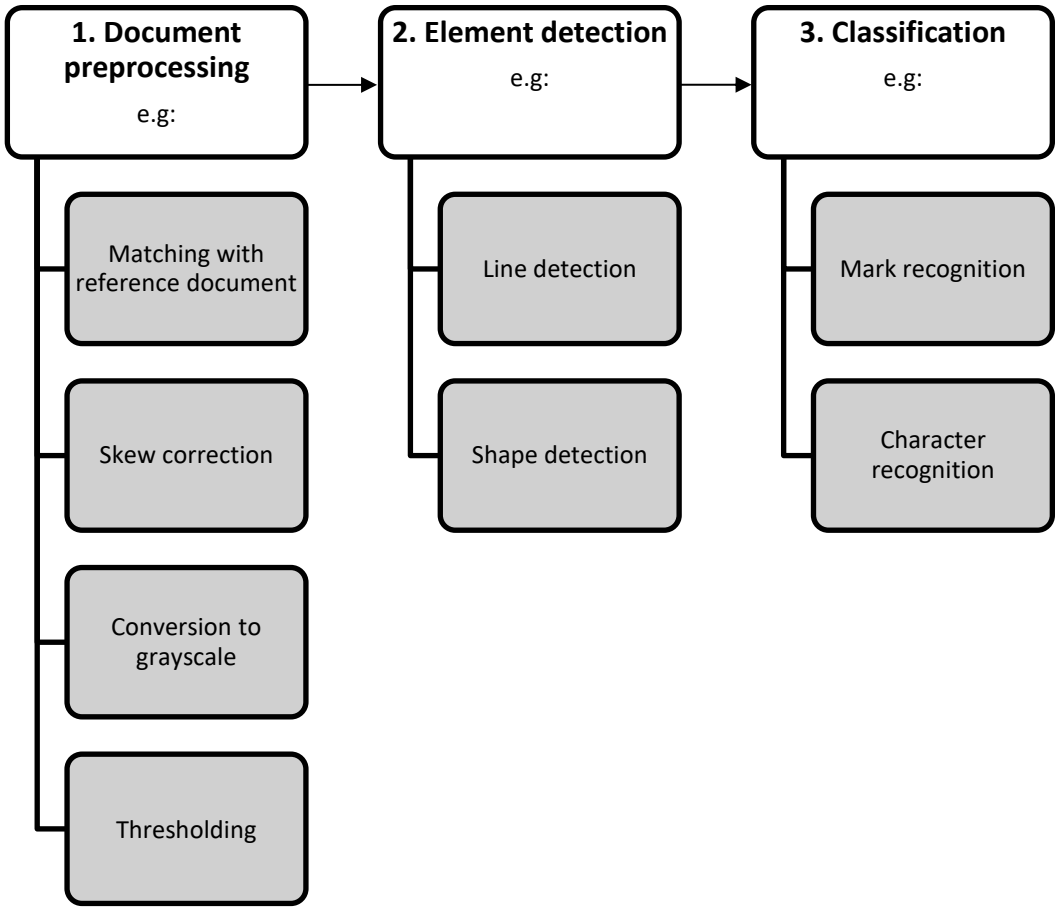


Figure 4. Common sequential workflow in computer vision systems,

2.2.4 The Visual Analog Scale, VAS

As there already exists OMR techniques for reading multiple choice questions, one possible solution for automatic reading of the PrePain questionnaires is to transform them to OMR sheets and use existing OMR solutions. However, what precludes that approach is the fact that the questionnaires contain continuous scales, VASs, that from a scientific perspective are not always replaceable by discrete scales.

A visual analog scale (VAS) is a validated psychometric measurement used in questionnaires to, inter alia, assess chronic and acute pain. A visual analog scale is a line, often 100 millimeters long, representing a continuous scale with two anchors, one in each end. The anchors usually have words marking opposite ends of the scale, e.g. “no pain”

and ”worst pain” (Reips and Funke, 2008; Delgado *et al.*, 2018). Scores are recorded by measuring the exact position of a handwritten mark on the line with a ruler, and hence reading data from paper based VASs is very time consuming. The practical concerns of reading VAS:s are alleviated with computerized versions (Reips and Funke, 2008) and there exists tools for creating and reading digital VASs (Reips and Funke, 2008; Marsh-Richard *et al.*, 2009)

Nonetheless, paper-based questionnaires are still used to a great extent. One reason for this is that response rates and adequacy of response rates are higher for paper based surveys (Nulty, 2008). A tool for automatic decoding of paper based VASs could improve efficiency and robustness in several research areas relying on collection of VAS data. Related research about automatic reading of questionnaires and similar documents such as exam answer sheets makes use of optical mark reading (OMR). OMR is used to read multiple choice questions, where answers are made by marking boxes or bubbles. However, OMR requires discrete scales or categorical answers, and research have shown that VASs have a number of advantages over discrete scales and the two formats are not always exchangeable (Reips and Funke, 2008). For example, a study from 2006 showed that there is a systematic difference between equally spaced online radio buttons and VASs and that they do not show linear correspondence (Funke and Reips, 2006). A major advantage of VAS is that VAS data is interval-scaled, while data from categorical scales is ordinal-scaled. This enables calculation of the arithmetic mean from VAS, data, while only median values can be extracted from discrete categorical scales. (Klimek *et al.*, 2017)

2.2.5 Hough Line Transform

As described above, VASs are represented by lines in the image. Accordingly, these must be recognized in the document before it is possible to recognize marks on the VASs. When looking for ways to detect lines in images, one eventually stumbles upon the Hough Line Transform or one of its variations. Originally patented in 1962, the technique has been covered in and extended upon in hundreds of research papers (Mukhopadhyay and Chaudhuri, 2015).

In OpenCV, two variants of the Hough Line Transform are available. The standard version is implemented as a function `HoughLines()` and is explained in a tutorial. In short, a 2D line can be expressed with polar coordinates as $r = x\,cos\,\theta + y\,sin\,\theta$. Here, (x, y) is a point belonging to the line we want to express and is chosen so that a perpendicular line can be drawn from this point to the origin. This perpendicular line has a length r, and an angle θ starting from the x-axis. Now, consider all possible lines passing through (x, y). Each of these lines can be expressed in the same way, but the values of r and θ will vary. If constraints $r > 0$ and $0 < \theta < 2\pi$ are imposed and all possible values of r and θ for lines passing through (x, y) are plotted as points in a plane r-θ we obtain a sinusoid in this new plane. If we pick a new point (x_1, y_1) belonging to the same line and do the same process, we will obtain a second sinusoid in the r-θ plane.

Since one of the possible lines passing through (x_1, y_1) is the line we started out with, these sinusoids will intersect at a point (r, θ) corresponding to that line. The Hough Line Transform in OpenCV uses this fact to scan for lines in a binary image. For each white pixel, the sinusoid of the line family passing through it are drawn in the r-θ plane. Since all white pixels belonging to the same line will yield a new intersection lines can be identified by setting a threshold for the least number of intersections (OpenCV, 2020f).

However, only obtaining r and θ gives no information of the length of the line, where it begins, or where it ends. The other variant available in OpenCV, implemented as `HoughLinesP()`, solves this problem and gives the extremes of the detected lines (OpenCV, 2020c). While OpenCV refers to this implementation as the Probabilistic Hough Line Transform (ibid.), their cited source instead calls it the Progressive Probabilistic Hough Transform and points out that it differs from the Probabilistic Hough Transform (Matas, Galambos and Kittler, 2000).

2.2.6 Skew Correction

When working with digitized documents, a skew angle is often introduced due to small misalignments during the scanning procedure, and correcting this skew is crucial in most document analysis systems (Boudraa, Hidouci and Michelucci, 2020). Horizontal elements, such as lines, will not be identified in systems that searches along the horizontal axis if the skew angle is too high (Yefeng, Huiping and Doermann, 2005). Boudraa, Hidouci and Michelucci (2020) describes and categorizes several existing methods that deal with this problem in documents where no ground truth is available. Projection profile analysis, nearest-neighbor, and Hough transform based methods among others are mentioned by the authors.

While perhaps not as common when using a document scanner, document images may have perspective distortion. This type of distortion can be described as some parts of the document being larger due to being closer to the capturing equipment (Fisteus, Pardo and García, 2013). Perspective distortion can be rectified by finding a perspective transformation between the distorted plane and the undistorted plane and applying it to the image. The perspective transformation, or homography matrix H, is a 3×3 matrix so that

$$s_i \begin{bmatrix} x_i' \\ y_i' \\ 1 \end{bmatrix} \sim H \begin{bmatrix} x_i \\ y_i \\ 1 \end{bmatrix}. \tag{4}$$

Here, s_i is a scale factor and H map points between the distorted and the undistorted plane. In OpenCV, the function `findHomography()` searches for H given a set of corresponding points in the two planes (OpenCV, 2020a).

To find corresponding points several algorithms exist. One such algorithm is ORB (Oriented FAST and Rotated BRIEF), which builds upon the FAST keypoint detector and

the BRIEF feature descriptor. In this context keypoints, or features, in an image refers to corners, blobs, edges, and similar objects detectable by a keypoint detector algorithm. The descriptor algorithm in turn creates descriptions of these keypoints. BRIEF is robust against lightning, blur, and perspective distortion. The Rotated BRIEF variation, which is the one used in ORB, is in turn robust against rotation. This means that given two images of the same motive, ORB can detect the same keypoints in both images and create descriptors that match even if the images are of different perspectives (Rublee *et al.*, 2011).

2.2.7 OMR Mark Detection

The PrePain questionnaire entails that the person places marks on VAS lines and, in the second version of the questionnaire, in checkboxes. These marking have to be decoded. Existing OMR mark detection techniques used for decoding marks in answer boxes and/or bubbles include adaptive x-mark matching (Chouvatut and Prathan, 2014), projection profiles (Sattayakawee, 2013), pixel counting (Haskins, 2015), finder patterns (Chai, 2016), thresholding (S, Atal and Arora, 2013) and machine learning models like CNNs and bag of visual words (Afifi and Hussain, 2017). The plethora of mark detection methods in combination with different preprocessing techniques indicate that there are few general solutions. Methods are often tailor-made for a type of document or input field, and many OMR systems make use of a sequential combination of techniques. (see e.g. Loke, Kasmiran and Haron, 2018). Loke et al. mention several factors affecting the performance of a mark detection technique, e.g. whether the respondents are trained or untrained when filling in a sheet, what pen or pencil is used, and presence of artifacts caused by scanners and printers (2018).

One of the simpler methods for mark detection is thresholding, described by (S, Atal and Arora, 2013). The authors present a system for designing, registering and evaluating forms. In this context, registration refers to handling the variance in rotation and translation of forms. This variance originates from manual error in aligning the documents during scanning, see Section 2.2.6. For their method to work, the position of all bubbles/boxes must be fixed to the same position in all scanned images before further processing of the forms (S, Atal and Arora, 2013). In this case, registration is accomplished by detecting square boxes in each corner of the form and first rotating and then translating the image. The answer boxes are then identified by detecting rectangular contours (rectangles containing groups of bubbles), and the relative position of the answer bubbles is known within these. To determine whether the bubble is filled or not, the average grayscale value of the smallest rectangular region bounding the answer bubble is calculated and compared to a threshold. The idea behind this is that the there is a significant difference between the average grayscale value of a filled bubble and that of an unfilled bubble. Completely black pixels have the minimum pixel value 0 and white pixels have the maximum value 255. A low average grayscale value thus indicates that a box is filled, while unfilled bubbles have a high grayscale average. If the average grayscale value, V_i, satisfies

$$Vi < Vmin + (Vmax - Vmin) * p. \quad (5)$$

it is considered filled (S, Atal and Arora, 2013). If V_i instead fulfills

$$Vi > Vmin + (Vmax - Vmin) * q \quad (6)$$

it is considered unfilled (ibid.). For each scanned image, $Vmin$ denotes the minimum average grayscale value and $Vmax$ denotes the maximum average grayscale value. The variables p and q are user-defined adaptive threshold factors where $0 < p < q < 1$ (ibid.).

These comparisons will not work as expected if all bubbles are filled or unfilled. To handle this case, a threshold parameter based on the average of $Vmin$ and $Vmax$ is compared against the absolute max and min (ibid.).

A problem discussed by the authors is that poorly erased marks are detected as marks (ibid). The method is more biased towards erroneously detecting erased filled in bubbles than erased cross marks (Loke, Kasmiran and Haron, 2018).

Another simple yet successful method used to detect marked "hotspots ", i.e. bubbles or boxes, is called pixel counting (ibid.). Haskins (2015) concludes that the pixel counting method performs better than other more sophisticated methods in terms of accuracy, sensitivity, specificity and agreement (Cohen's kappa). Haskins proposed method begins much like that of s. Atal and Arora (2013) with de-skewing and resizing of each scanned image by comparing it to a memorandum with reference squares in each corner. Thereafter, each hotspot is analyzed with a padding of 11 pixels on each side. The extra padding is kept partly to counter slight variances that remain despite de-skewing and rotation, and partly to keep information about the area around the box, e.g. whether it has been crossed over. For each pixel in the remaining hotspot, the average RGB value is calculated, and compared to an empirically derived threshold. Each pixel is then flagged as marked or un-marked and the percentage of marked pixels in each hotspot is in turn compared to a threshold to determine whether it is selected or not. Haskins (2015) use 200 as threshold for each pixel to be considered marked and requires 20% of the pixels within a hotspot to be marked in order to classify it as selected. However, this method cannot distinguish between selected and de-selected hotspots, and the author suggests that a combination of pixel counting and another method better at detecting deselections would be needed to achieve this behavior (ibid.).

A similar but slightly different approach to differentiate marks is based on the additive sum of all pixel values after adaptive thresholding. This method tends to detect more vague marks and lighter pixels (Loke, Kasmiran and Haron, 2018).

2.2.8 Shape Detection and Contour Approximation

Geometric shapes in images, such as rectangles, triangles and circles, can be detected by identifying contours and approximating polylines to these. A contour can be described as

a curve joining all continuous points along a boundary having the same color or intensity (Mordvintsev and Abid, 2014). In 1985, Suzuki and Abe proposed two algorithms for extracting the topological structure of images. Contours, or borders, are found between background, connected components and holes, see Figure 5. Contours can be identified using the built in function `findContours()` in OpenCV. The function retrieves contours from a binary image (*OpenCV: Structural Analysis and Shape Descriptors*, 2018), and is based on algorithms presented by (Suzuki and Abe, 1985). Depending on which "contour retrieval mode" `findContours()` is called with, it returns either all contours, only the outer contours, or contours organized in a hierarchy of nested contours (*OpenCV: Structural Analysis and Shape Descriptors*, 2018). Each individual contour is represented as an array with the object's boundary points (x, y) coordinates (Mordvintsev and Abid, 2014).

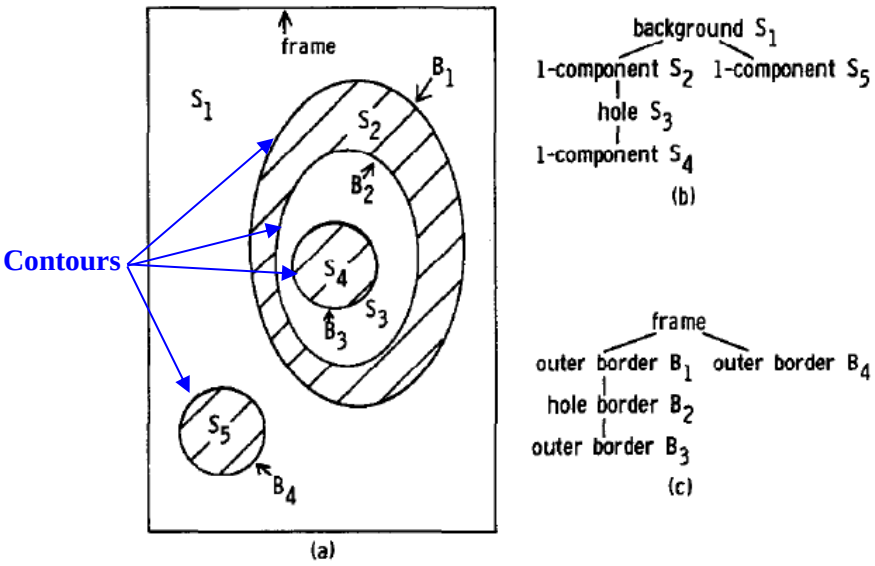


Figure 5. Surroundness among connected components (b) and among borders (c). Adapted from Suzuki and Abe (1985).

To further identify the shape of the retrieved contours, they can be approximated using the Douglas-Peucker algorithm. The Douglas-Peucker algorithm, or Ramer-Douglas-Peucker algorithm, was developed within the field of cartography, and is an algorithm for reducing the number of points in a polyline/curve by approximating it to a similar curve with fewer points (Douglas and Peucker, 1973). The algorithm is initiated by drawing a line between the first and last point, these points are always kept. If the Hausdorff distance to the point farthest away from this line is greater than a chosen threshold ε, that point is kept. The algorithm then recursively calls itself with the first point and the farthest point, and then with the farthest point and the last point. In the case of approximating contours, the interpretation of the parameter ε is the maximum distance allowed between the actual contour and the approximated contour. In general, smaller ε give more accurate approximations, but require more points to describe the approximated curve. When the recursion is completed, i.e. no farthest point is farther away from the approximated line

than ε, only the points that have been marked as kept are outputted. The OpenCV function `approxPolyDP()` is used to implement the Douglas-Peucker algorithm (*OpenCV: Structural Analysis and Shape Descriptors*, 2018).

With an appropriate choice of ε, the shape of geometrical objects can be determined by calling `approxPolyDP()` and analyzing the number of edges required to approximate a contour. If three edges are needed, it is a triangle, if four, it is a quadrilateral, if five, it is a pentagon, etc. This works even if the object's shape is "imperfect".

2.2.9 Handwritten Digit Recognition with CNNs

Aside from detecting lines, shapes and marks in the questionnaires, a numeric ID written by hand is a part of the second version of the questionnaire. Recognition of handwritten digits is included in the field of ICR. Convolutional neural networks, also called CNN: s or ConvNets, have shown great performance in image classification, object detection, character recognition etc. This is due to their ability to preserve spatial relations in the input data. A multitude of CNN architectures have been proposed for handwritten digit recognition, and CNN: s tend to outperform other classification algorithms such as K Nearest Neighbors, Support Vector Machines, and fully connected neural networks with backpropagation (Liu, Wei and Meng, 2020). The use of elastic distortion in augmenting the training data is found to be important in order to achieve low error rates, and deep nets perform better than shallow ones. (Deng, 2012) A recently proposed pure CNN architecture achieves a 99.87% accuracy on the MNIST data (Ahlawat *et al.*, 2020). Another state-of-the-art CNN, EnsNet, which makes use of ensemble learning, achieves an accuracy of 99.84%. (Hirata and Takahashi, 2020)

2.2.10 The MNIST Database

Training of machine learning models require large amounts of data. The MNIST database, (Modified National Institute of Standards and Technology database) is a widely used large database containing handwritten digits. The MNIST database is freely available and has a training set of 60,000 images and a test set of 10,000 images (Lecun *et al.*, 1998). It is commonly used to assess the relative performance achieved by different machine learning algorithms and preprocessing techniques (Deng, 2012).

Each MNIST example is a 28×28 grayscale image in which the digit is size normalized to fit within an inner 20×20 pixel box. The inner 20×20 pixel box containing the digit is centered in the larger 28×28 image by its pixel mass center (Lecun *et al.*, 1998).

2.3 Quality Control of MRI data

The PrePain project involves collecting anatomical and functional MRI. MRI are 3D or 4D images of the brain that are used to infer brain activity. MRI images also contain noise which can dramatically affect the data. An example is when a subject moves considerably during a scanning sequence, which can be problematic for subsequent analyses (Power *et al.*, 2012; Van Dijk, Sabuncu and Buckner, 2012; Power, Schlaggar and Petersen, 2015). Thus, identifying subjects with high amount of noise is an important issue in MRI analyses.

MRIQC (Magnetic Resonance Imaging Quality Control tool) is an open source project developed by the Poldrack Lab at Stanford University. MRIQC extracts IQMs (Image Quality Metrics) from MRI images, and is a contribution to the work of automating quality assessments of neuroimaging data. The main motivation behind quality assessment is that low quality MRI data may lead to faulty conclusions when analyzing neuroimaging data (Esteban *et al.*, 2017). MRI images are seldomly artifact-free, and artifacts can be patient related, signal processing dependent as well as hardware (machine) related. (Erasmus *et al.*, 2004)

Exclusion of low-quality images has traditionally been done through visual inspection by experts, which is both time-consuming and affected by inter-rater differences. In addition to this, the trend towards large-scale studies with large datasets from multiple sites has increased the need for reliable, robust and efficient automated quality control (Esteban *et al.*, 2017)

MRIQC returns a set of image quality metrics (IQMs) extracted from MRI images, which are grouped in four broad categories:

- Measures based on noise measurements
- Measures based on information theory
- Measures targeting specific artifacts
- Other measures

See Table 13 in appendix C and Table 14 in appendix D for complete lists of the IQM features. The IQMs will be further described in Section 4.1.

In addition to introducing MRIQC in their article from 2017, Esteban et al. evaluate two families of binary classifiers' ability to predict the quality (exclude/accept) of new MRI data from unseen sites. Labeled data was retrieved by letting two experts visually inspect MRI data and assign quality labels (exclude/doubtful/accept). The labeled MRIQC data was then used to train classifiers through supervised learning. An accuracy of 76% was at best obtained by a random forest classifier, when evaluated on data from an unseen site. The authors conclude that the classification is susceptible to what they call "site-effects" and that the trained classifier does not generalize well to data from new sites. The site-effects are explained by the fact that IQMs are highly correlated with the site the MRI

image data comes from, and that this accounts for much of the variability in the data (Esteban *et al.*, 2017).

In *Improving out-of-sample prediction of quality of MRIQC,* the site-effects in the IQMs are analyzed further and confirmed by t-SNE plots (Esteban, Poldrack and Gorgolewski, 2018). The labeled data used in the 2017 experiment showed a very high inter-rater variability, which the authors suspect to be caused by annotation errors. A label binarization step in the 2017 experiment in which all data points labeled "doubtful" by the experts were mapped as "accept" in the binary classification was therefore revised in 2018, which lead to an accuracy improvement of 10% (Esteban, Poldrack and Gorgolewski, 2018). T-SNE projections show that IQMs are highly structured around their site of origin, even when the features are site-wise normalized. What is more, the t-SNE analysis does not reveal any clear relation between the IQM features and manual quality ratings, which according to the authors suggest that 1) there is a need for better features that are more representative of quality, and 2) the problem of automated quality classification is not reliably solved with simple models (Esteban, Poldrack and Gorgolewski, 2018).

Unless the user actively opts out, anonymized quality metrics (IQMs) are uploaded to a publicly accessible web server each time MRIQC is run (Esteban *et al.*, 2019). IQMs extracted from MRI images are collected in the crowdsourced database MRIQC Web-API[1]. The intention is for it to be used for training human experts as well as machine learning algorithms (ibid.). The purpose of the data collection is to "build normal distributions for improved outlier detection" (*Running mriqc — mriqc 0.1 documentation*, no date). A MRIQC user also has the opportunity to rate the image quality and motivate the rating with descriptive words. If this is done, and the user does not actively oppose, the rating is also uploaded to the MRIQC database. According to Esteban et al, the ultimate goal of the database is to develop a fully automated quality control tools that performs better than human experts in identifying subpar images. The focus of quality control (QC) is to identify outliers and flag images that, due to their poor quality, may pose a threat to downstream analysis (Esteban *et al.*, 2019).

In functional Magnetic Resonance Imaging (fMRI) studies, the standard way to generate images is by using Blood Oxygenation Level Dependent (BOLD) contrast (Shah *et al.*, 2010). While MRI studies brain anatomy, fMRI studies brain function. BOLD signals are an indirect measure of functional brain activity. As a complement, a high-resolution T1-weighted anatomical image is often used to map activations onto. In addition to T1-weighted images, usually only referred to as T1, there are T2-weighted images. Both T1 and T2 are basic pulse sequences in MRI.

[1] Available at mriqc.nimh.nih.gov

2.3.1 Anomaly Detection

As stated by Esteban et. al (2019), the purpose of collecting anonymized MRIQC data in a database is to develop automated quality control, with focus on identifying and flagging outliers. This is referred to as *anomaly detection.* Anomaly detection, or outlier detection, is the task of identifying observations that differ significantly from the majority of the data. When the task is to decide whether a new observation is an outlier or not, the term novelty detection is commonly used. Anomaly detection techniques can be divided into three major groups: supervised, unsupervised and semi-supervised. Supervised learning techniques require labeled data (anomaly/normal), while unsupervised models learn from unlabeled data. In semi-supervised techniques, only some of the data is labeled, an example is when all training data is assumed to belong to the normal class.

Several definitions of what an outlier is exists. Hawkins defines an outlier as "an observation which deviates so much from other observations as to arouse suspicions that it was generated by a different mechanism (Hawkins, 1980). Barnett and Lewis defines it as "an observation (or subset of observations) which appears to be inconsistent with the remainder of that set of data" (Barnett and Lewis, 1994).

Outliers may arise from data entry errors, experimental errors, data processing errors, or variability in measurement. However, they need not be caused by errors, some are simply natural novelties in data. There are different types of outliers, e.g. single point outliers, contextual outliers and collective outliers. Outliers in n-dimensional space are referred to as multivariate outliers, while univariate outliers are extreme data points in the distribution of one variable.

2.3.2 Isolation Forest

An unsupervised machine learning algorithm for anomaly detection is isolation forest, also called *iForest*. It was first introduced in 2008 (Liu, Ting and Zhou, 2008), and further evaluated in 2012 (Liu, Ting and Zhou, 2012). Isolation forest differs from many other statistical-based, classification-based and clustering-based anomaly detection techniques in that it does not construct a profile of normal instances. Instead, it is based on the idea of anomalies being few and different, and thus susceptible to *isolation.* Liu et. al define the term isolation in this context as "separating an instance from the rest of the instances"(Liu, Ting and Zhou, 2008). Isolation forest is a tree-based ensemble method. Since outliers are assumed to be few and different, an inherent characteristic is that when a tree structure is constructed to isolate every single instance in a data set, outliers are closer to the root of the tree while normal observations are found at a greater depth. Anomalies are simply easier to isolate. See Figure 6. An anomalous instance requires fewer random partitions to be separated than a normal one and hence has a shorter average path length when building an ensemble of isolation trees, iTrees.

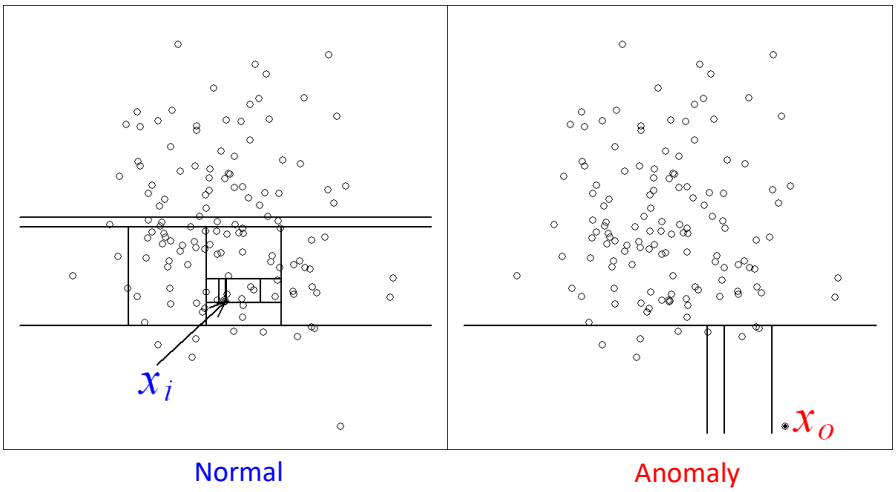


Figure 6. Isolating a normal point, x_i, requires twelve random partitions (left). Isolating an anomalous point, x_0, requires fewer random partitions, in this case four (right). Adapted from Liu, Ting and Zhou (2008)

An isolation tree is defined such that a data sample $X = \{x1, \ldots, xn\}$ of n instances from a d-variate distribution is recursively divided by selecting an attribute q and a split value p. The recursion progresses until i) the tree reaches a height limit, ii) $|X| = 1$, or iii) all data points in X have the same value. Each node in an isolation tree has exactly two or zero child nodes. This kind of trees are referred to as proper binary trees

Since isolation trees are proper binary trees, the path length $h(x)$ of a point x is equal to the number of edges x traverses from the root node until termination at an external node. The average path length is calculated in the same way as for unsuccessful search in binary search trees (BSTs). The average path length $c(n)$ of unsuccessful search in a BST is given by the equation

$$\mathrm{c(n)} = 2\mathrm{H(n} - 1) - \frac{2(n-1)}{n}, \tag{7}$$

where $H(i)$ is estimated by $ln(i) + \gamma$, where γ is the Euler–Mascheroni constant[2]. The anomaly score, which is a rank of degree of anomaly of an instance x, is defined as

$$\mathrm{s(x, n)} = 2^{-\frac{\mathrm{E(h(x))}}{\mathrm{c(n)}}}, \tag{8}$$

[2] $\gamma = 0.5772156649$

where $E(h(x))$ is the average path length $h(x)$ from a collection of n instances. From this follows that

• when $\mathrm{E}(\mathrm{h}(\mathrm{x})) -> \mathrm{c}(\mathrm{n}),\ \mathrm{s} -> 0.5$;

• when $\mathrm{E}(\mathrm{h}(\mathrm{x})) -> 0,\ \mathrm{s} -> 1$;

• and when $\mathrm{E}(\mathrm{h}(\mathrm{x})) -> \mathrm{n} - 1, \mathrm{s} -> 0$.

This means that:

i) if instances have an anomaly score s close to 1, they are most definitely anomalies;

ii) if s is much smaller than 0.5, the instances can be regarded as normal and

iii) if $s \approx 0.5$ for all instances, there is no distinct anomaly in the sample (Liu, Ting and Zhou, 2008).

A problem for many anomaly detection techniques, and machine learning models in general, is that they suffer from "the curse of dimensionality". One manifestation of this is that since data becomes sparse in high-dimensional space, distance is no longer a meaningful measure. According to Liu et al., iForest is not exempted from the curse of dimensionality, and it is therefore recommended to select an attribute subspace to reduce the dimensionality before constructing iTrees. Two other common problems in anomaly detection is swamping and masking. Isolation forest overcomes these effects by sub-sampling. In fact, isolation forest work best on small sample sizes. Isolation forests are therefore trained on sub-samples from random selection of instances without replacement. In the testing stage, test instances are passed through isolation trees in an iForest and the anomaly score s, calculated with Eq. 8, is returned as a measure of degree of anomaly (Liu, Ting and Zhou, 2008).

Difficulties in earlier attempts to use supervised machine learning models on labeled MRIQC data to train a classifier to predict quality from MRIQC data indicate that it is not a simple task, and that the relationship between IQM features and actual quality is complex. Despite this, there is a desire among researchers to use MRIQC data to get an indication of the quality of the image, and above all, to identify images that are of such low quality that they can harm further analysis of the data. If it is assumed that 1) images with substandard quality are exceptions, and not the normal case; and 2) these abnormalities are reflected in the MRIQC IQMs, outlier detection through Isolation Forest appears to be a viable way to identify these anomalous data points.

3. Automatic Decoding of VAS Questionnaires

In chapter 3 the methodology and results from the questionnaire decoding subtask is presented. Section 3.1 is a step by step description of the problems and the proposed solution for each problem. In section 3.2 the results are presented.

3.1 Methodology and Data

In this section, the methods used to decode the questionnaires are presented. The PrePain questionnaires that were used in practice at the start of the project were originally not designed for machine reading. Parts of the questionnaire design complicated the automatic decoding. For example, the fact that the VAS lines were dashed contributed to difficulties for precise line detection. Furthermore, the layout of the binary answers was not compatible with previous work within OMR. Due to this, the results of the first attempts of automatic decoding of the questionnaires did not suffice.

In this situation, two alternatives were considered:

1. Improve the methods.

2. Improve the input data.

Given that previous research have shown great performance in OMR detection of answer boxes, and that relatively small changes in the questionnaires were required to adapt them to computer vision methods, it was decided to go for the second option; to improve the questionnaires.

In addition to aiding the task of automatic decoding, as the data recording would now be handled automatically, a need for identifying each questionnaire with an ID emerged which lead to the addition of an ID field on the first page of the questionnaire. This broadened the scope of the project, as it required implementation of a method for handwritten character recognition. All changes in the layout of the questionnaire are described further in the following sections and summarized in Section 3.2.1. Both the first version of the questionnaire and our proposed design is found in Appendix A and B.

The *VASReader* system follows a sequential workflow and consists of five main modules, see Figure 7 below. The methods are based on the existence of a reference questionnaire for alignment and skew correction. The empty reference questionnaire is used for all types of object detection. The recognition of VAS marks, checkbox marks, and handwritten digits, is then performed in the skew corrected filled-in questionnaires, using the coordinates retrieved from object detection in the reference document. In Figure 7, the methods that are applied to the reference document are marked by an R, and methods that are applied to a filled in questionnaire are marked by an Q. Each step of the *VASReader* system will be described further in the following sections.

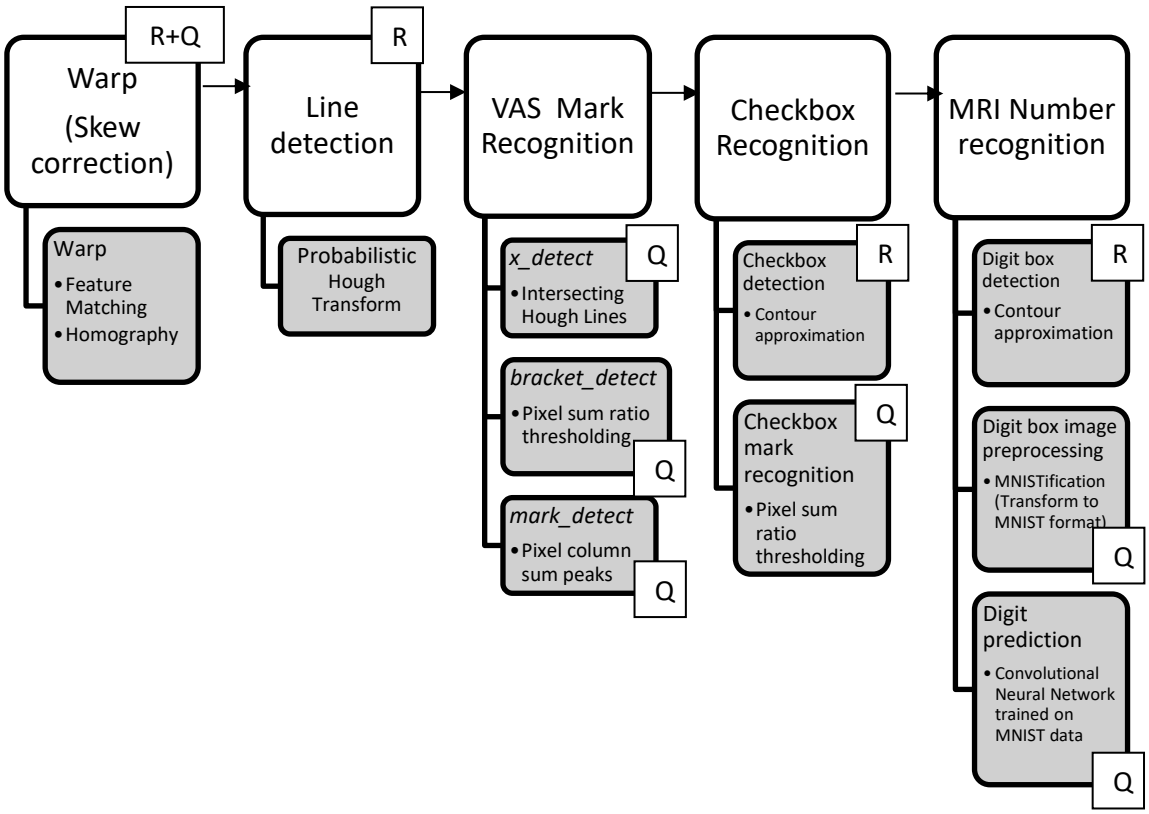


Figure 7. Overview of the five sequential steps of VASReader. The main methods used in each step are summarized in the gray boxes. Methods applied to the reference document are marked with an R, while methods applied to the filled-in questionnaire images are marked with a Q

3.1.1 PrePain Questionnaire Data

When developing *VASReader*, 40 anonymous filled out questionnaires (first version) were provided by KI as example data. From these it became evident that patients do not necessarily fill out the VAS questions with crosses even if instructed, many different mark variations were observed. Furthermore, not all the filled-out questionnaires originated from the same original document. VAS line lengths varied a bit between different versions but were otherwise largely identical.

In the second version of the questionnaire, all lines were solid and of equal length, checkboxes were used for binary questions, and a field for filling in a questionnaire ID (MRI number) was included in the top left corner of the first page. In addition to that, the second version was identical to the first one.

To evaluate the performance of *VASReader*, 29 filled-in questionnaires of the second version were collected and manually measured with a ruler. It should be mentioned that

this test data, in contrast to the first 40 questionnaires that were provided by KI, were not filled in by actual study participants. The data was thus "fabricated", but to get a variance in handwritings, pens used etc., a diverse group of persons were asked to fill in questionnaires. Some questionnaires were also on purpose filled-out in ways observed from the first set of data from real participants. Variations in mark shapes (circles, crosses, lines), different handwritings, sizes and cases of multiple answers were included.

3.1.2 Skew Correction

When first conceptualizing how to get a computer to find the correct elements in a scanned document, the decision to try to use the original document as a template was made. The reasoning behind this was that it would be comparably easy to test and evaluate different element detection methods on an empty document, and then be able to use information about detected elements when searching in a scanned document. However, to use that information the elements would have to be in the same place in the scanned document. As scanned documents were assumed to always have a small skew angle and be slightly misaligned, both these problems would have to be solved to be able to use information from the template. Any conventional skew correction technique would only solve the skew angle problem. Therefore, a feature matching and homography technique based on the core principles described in Section 2.2.6 was explored instead.

OpenCV provides a tutorial to find objects in images. In the example, a specific food box is identified in a stack of different boxes by using a clear picture of the box design as a reference. This is done by using feature detection in both images, matching the features, and finding the homography matrix to in turn find the object location adjusted for perspective distortion (OpenCV, 2020d). This was used as a guideline for how to perform feature matching. Initially, a distorted version of the PrePain questionnaire document was created using Microsoft Paint. By using the code from the tutorial, many features could be matched between the original questionnaire and the distorted version. The detected features consisted mainly of edges and letter serifs.

To correct for skew and distortion the homography matrix transformation can be applied to the distorted image using the OpenCV function `warpPerspective()` (OpenCV, 2020e). Using this technique in conjunction with the procedure described above, the resulting output is an image with the following properties:

- The output image is scaled to the same resolution and aspect as the original template image.
- No skew angle is present in the output image.
- All objects, such as lines as text, are aligned with the original template image with almost pixel-perfect precision.
- Perspective distortion is eliminated, although some artefacts remain in a heavily distorted input image.

Given the previous assumption that scanned questionnaires would be misaligned and have a slight skew angle, this combination of properties is very advantageous as all problems are addressed.

The OpenCV tutorial uses the patented feature detector and descriptor SIFT, which is still available for testing purposes (OpenCV, 2020c). A faster alternative to SIFT is ORB (Rublee *et al.*, 2011), previously discussed in section 2.2.6. However, when using ORB with the PrePain questionnaire only a small number of features is detected by default, leading to an unusable result. To overcome this, the maximum number of features detected by ORB had to be increased to an excessive amount. Using 50 000 as the maximum number, the same overall result could be achieved as with SIFT. The performance however decreased.

3.1.3 Line Detection

Finding lines in the PrePain questionnaire was initially straightforward. By experimenting with the `HoughLinesP()` function in OpenCV (OpenCV, 2020c), previously described in Section 2.2.5, all the VAS lines could easily be detected in an empty questionnaire. The main issue however was that too many lines were detected. The function can be tuned to allow for gaps in a line (ibid.), which was necessary as the VAS lines were dotted. This caused another problem; text serifs would be tied together into lines where none should be detected. A second problem was that the square brackets indicating the endpoints of each line were included in the line, possibly making each line longer than intended. A third problem was that for each VAS line five or more lines were detected, one for each pixel in the line width.

To solve the main problem with too many lines overall, predefined areas were created by manually zooming in on textless areas were lines should be detected. As lines were always appearing in the same place due to the skew correction step described above, predefined coordinates could be used for zooming. The third problem was subsequently solved by creating bins of adjacent lines and always picking the middle one out of each bin.

The second problem proved a bigger issue. Brackets were included in most detected lines, but not all. Picking the middle line thus led to varying line lengths. To overcome this problem the brackets had to be excluded altogether. By using a kernel of some shape the morphological operations erode and dilate can shrink or grow elements of an image based on the kernel shape (OpenCV, 2020b). Eroding, and then dilating, around a horizontal line shaped kernel removed all vertical elements in the line areas. Removing brackets resulted in all detected lines in each bin being the same length.

When tuning the line detection algorithm on filled out questionnaires it became evident that pencil strokes could prolong the detected line, a mark close to or at the ends of a line was interpreted as being part of the line. As this could impact the measurement it was decided that line coordinates should be obtained from the reference document, rather than the filled-out questionnaires. This approach relies heavily on the skew correction step

working properly as lines need to appear in the same pixel locations across different scanned questionnaires.

3.1.4 VAS Mark Detection

When lines have been correctly recognized, the subsequent step is to detect handmade marks on the VAS lines. Several factors contribute to the complexity of this task. To begin with, there is a great variance in how marks are made. The solution for VAS mark detection had to consider a number of challenges, which we first list.

The first challenge was that some participants mark their answers by X-shaped crosses, others by horizontal lines or circles. Some marks even resemble the Greek letter alpha (deriving from making a cross without lifting the pen). The size of the marks also varies. Moreover, different pens and pencils are used, with different thicknesses and intensity. X-shaped crosses are seldom positioned with the intersection between the two "X-legs" exactly at the line, and thus intersects the line in two points. With this comes challenges such as how to distinguish a cross intersecting the line at two points from two separate horizontal marks.

The second challenge that emerged was whether to record the first intersection, the second, the mean between the two, or the middle of the cross. It also became evident that many marks in the extremes of the VAS are positioned outside of the line, in the area between the line and the brackets. It was thus not sufficient to only search for marks on the line, but necessary to also be able to detect marks to the left and right of the line.

The third challenge was how to handle multiple marks on the same VAS, which occurs when participants regret their first answer and instead of erasing it make a new, sometimes stronger, mark. All in all, the above described challenges helped shape the requirements on the mark detection algorithm. The requirements specification is presented below in Table 2.

Table 2. Requirements on the VAS mark detection algorithm

The mark detection module of VASReader is expected to:
▪ Detect marks of varying shapes and sizes
▪ Flag when the line is ambiguously marked, e.g. if there is more than one mark on the line
▪ Detect marks to the left and right of the line
▪ If possible, detect the intersection of an X-mark even if it not placed exactly on the line, else, detect the first intersection with the line.
▪ Detect if the line is unmarked.

To meet the challenges, the mark detection module is built as a combination of three main methods that, when combined, form a robust and flexible mark detection algorithm. The three methods complement each other as each method is specialized in handling a subset of the cases described above. Furthermore, they are internally prioritized in a thought out way. See Figure 8. Method one, *x_detect*, is specialized in finding x-shaped marks. Method two, *bracket_detect*, is specialized in finding marks close to the brackets. Method three, *mark_detect*, is the most general method, detecting marks of any shape intersecting with the VAS line. The third method, *mark_detect,* also has built in functionality for flagging ambiguously marked lines. If none of these three methods is able to detect a mark, the line is flagged as unmarked. Each method will be presented further in the following three sections.

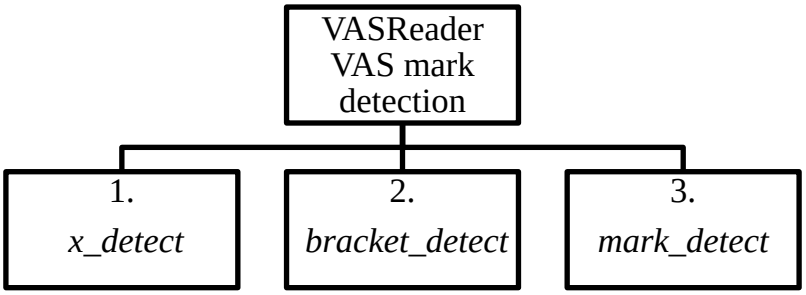


Figure 8. VASReader's VAS mark detection recognition module is composed by three separate methods.

3.1.5 VAS Mark Detection Method 1: *x_detect*

In the case where a participant puts an x-shaped mark it proved quite common that the center point appeared slightly above the VAS line, while only the legs of the x-shaped mark touched the line. It was assumed that the center point, or internal intersection point, was the intended mark location in all cases. The *x_detect* method was devised to identify the center point of x-shaped marks and project it onto the VAS line.

First, a rectangular search area around each line is defined. A line search is then performed with `HoughLinesP()`, resulting in a large collection of pixel-wide lines making up the x. As it is possible to fit intersecting pixel-wide lines in a single pencil stroke, lines are separated into two bins based on their angle sign, resulting in one bin for each leg. To further avoid lines stemming from the same pencil stroke being assigned to different bins, lines with angles close to zero or ± 90 are discarded. All lines in each bin are then compared to all lines of the opposite bin, and intersection points are stored. Finally, the mean intersection point is compared with the median intersection point. If these are sufficiently close, the horizontal x value of the mean intersection point is chosen as the

mark location. If they are not sufficiently close, the mark is flagged as being ambiguous and no answer is given.

In general, *x_detect* is very accurate for clearly written x-shaped marks. It does not identify marks of other shapes. However, it often identifies a mark even if it is crossed over, potentially giving a false positive. If a crossed over mark is replaced with a new mark, no answer will be given and the ambiguous flag will be set.

3.1.6 VAS Mark Detection Method 2: *bracket_detect*

Participants who want to mark an extreme value in a VAS, i.e. 0 or 100 mm, tend to place their mark on or near the side bracket. It is consequently necessary not only to search for marks on the actual line, but also to the left and right of it. The technique used for detection of marks close to the brackets is very similar to the method used to classify a checkbox as marked or unmarked, see Section 3.1.9. Since the line coordinates are known, a rectangular area containing the bracket and its closest surrounding area is sliced from the skew corrected image to the right and left of each line. This image is inverted and binarized as described in Eq. 3. The resulting images are seen in Figure 9. The ratio between the two bracket images' pixel sums is thereafter compared to a threshold. The comparison determines whether any of them is sufficiently enough stronger marked than the other. See Figure 10.

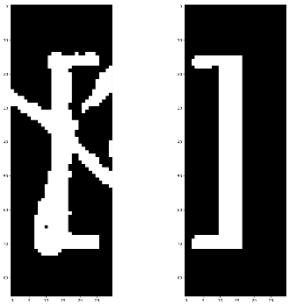


Figure 9. A marked bracket (left) and an unmarked bracket (right)

Figure 10. Two marks recognized by bracket_detect (in green)

3.1.7 VAS Mark Detection Method 3: *mark_detect*

The third method, *mark_detect*, searches for marks by finding peaks in column pixel sums in the areas above and. below the line respectively. The intuition behind the method is that when there is a mark intersecting the line, the sum of pixel intensities above and below that intersection is going to be significantly higher than it is elsewhere above and

below the line. Figure 11 is an example of a mark, and in Figure 12, the pixel intensity column sums below and above the line are shown in two graphs. As can be seen, there are clear peaks in column sums where the line has been marked. If there are peaks both above and below the line, the mark's position on the line is estimated as the mean between the first peak in the area above the line, and the first peak in the area below the line. In contrast to *x_detect*, this gives the position of the first intersection with the line, and not the intersection between the cross' two lines in the case of an x-shaped mark.

Figure 11. A mark on the line recognized by mark_detect (in red)

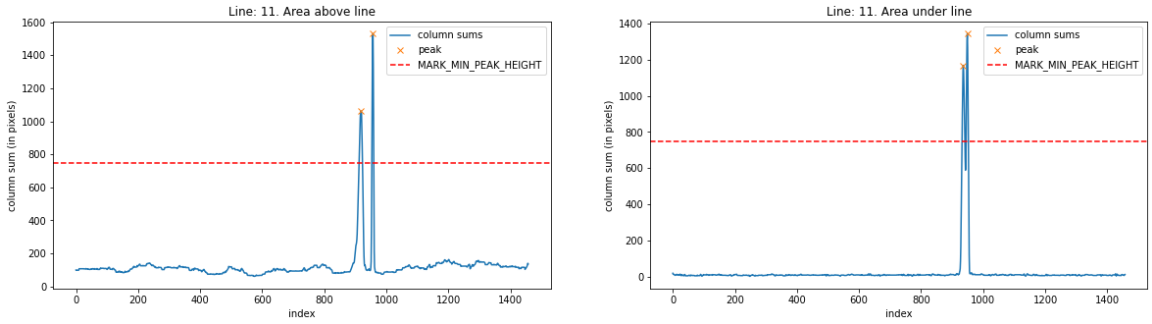


Figure 12. Plot of column sums of pixel intensities above and below the line shown in Figure 11. The peaks are shown by an x

The method also reacts on ambiguous markings, e.g when there are multiple peaks with a distance greater than a set threshold *allowed_distance*. An example can be seen in Figure 13 below, where the mark intersects the line in two places. The corresponding columns sum plots are seen in Figure 14. The peaks have a significant distance in x-direction. It should be noted that the cross intersection of this mark is still recognized by *x_detect*, but the line is flagged as ambiguously marked by *mark_detect*.

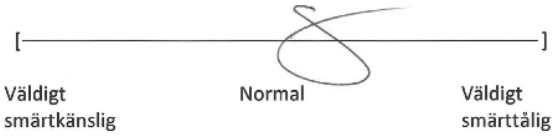


Figure 13. An ambiguous mark intersecting the lines in two places

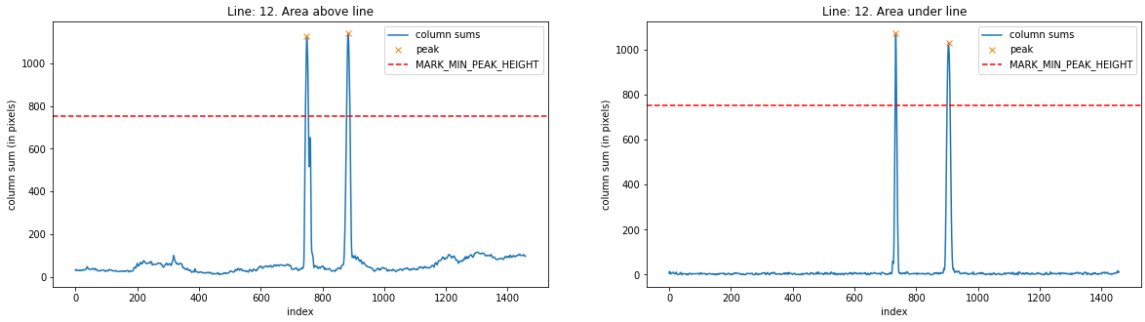


Figure 14. Plot of column sums of pixel intensities above and below the ambiguously marked line shown in Figure 13 above.

3.1.8 VAS Mark Detection Workflow

As all methods have different characteristics, strengths, and weaknesses, a decision mechanism is used to decide on a final answer. First, all three methods each provide a mark location estimate and a flag out of three possible states: *ok*, *no mark*, or *ambiguous*. No estimate is given if no mark was found. The final mark location estimate and its flag are given from a priority list and a set of rules:

- The final mark location estimate is given if at least one of the methods report an estimate with the *ok* flag. The final answer flag is also initially set to *ok* in this case.
- Consequently, if no method provided an answer with the *ok* flag, the final answer estimate is not given, and the final answer flag is set to *no mark.*
- Which estimate to choose, if there are more than one with the *ok* flag, is decided from the priority order: *x_detect* comes first, *bracket_detect* comes second, and *mark_detect* comes last.
- If at least one answer is flagged as *ambiguous*, the final answer flag is changed to *ambiguous* regardless if an estimate is given.

The reasoning behind the priority order stems from the different characteristics of the detection methods observed during development. In case of x-shaped or alpha shaped marks, *x_detect* generally outperforms *mark_detect* as the latter will always pick the first intersection with the VAS line, rather than the internal intersection at the center. For any mark shape located close to the brackets, *bracket_detect* correctly identifies which extreme to choose. For line shaped marks, *mark_detect* generally provides a good estimate. Furthermore, *x_detect* and *bracket_detect* should in theory not detect marks that are more accurately estimated by subsequent methods.

Non-extreme circle shaped marks or crossed over marks replaced with a new mark of any shape are not well handled by any of these methods. By making use of flags, it was decided to still provide a guess estimate where possible. An *ambiguous* flag indicates that human interaction is required, while still not requiring manual measurement if the guess is sufficient.

3.1.9 Checkbox Detection and Checkbox Mark Recognition

In addition to the VASs, the PrePain questionnaire contains five questions with binary answers (JA/NEJ), (Eng. YES/NO). One question is positioned at the top of the first page, and the remaining four questions are grouped together after the first VAS section. See Figure 15 and 16. In the first version of the questionnaire, used in practice at the start of this project, these were designed to be marked by being circled, see Figure 17. Initially, a technique for recognizing the answers from the binary questions by comparing pixel sums in the areas around the two answer alternatives was implemented but did not show satisfactory results. It became evident that there was a variance in how the answers were marked. Sometimes the answer was marked with a circle, sometimes with a cross, and sometimes by underlining. See Figure 18 for different variations. The JA and NEJ were placed close to each other, making it difficult in some cases to discern which one was marked, as circular marks could encircle one alternative and cross the other. In addition to this, two of the "JA NEJ" areas were positioned with short distance in y-direction to each other, and marks belonging to one question sometimes overlapped the marks of the questions above or below. See Figure 19 for an example.

Smärta just nu

- Har du smärta någonstans I kroppen just nu? JA NEJ
  Om JA, var? Vänligen sätt ett tydligt kryss och välj intensitet:

Figure 15. One of the five binary questions in the first version of the main questionnaire.

- Lider du just nu av långvarig smärta? JA NEJ
- Har du tidigare haft perioder med mycket smärta i kroppen? JA NEJ
- Har du någon i din nära familj som lider av långvarig smärta? JA NEJ
- Är du orolig för att drabbas av långvarig smärta? JA NEJ

Figure 16. Four of the five binary questions in the first version of the main questionnaire.

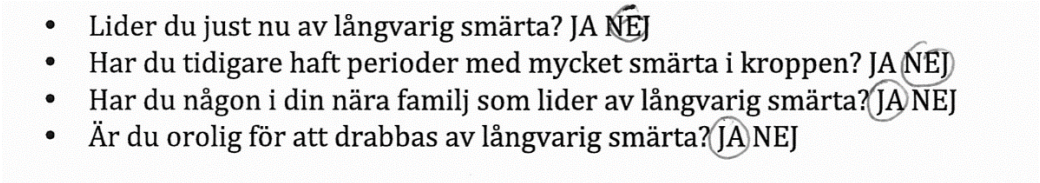


Figure 17. Marked answers in the first version of the main questionnaire.

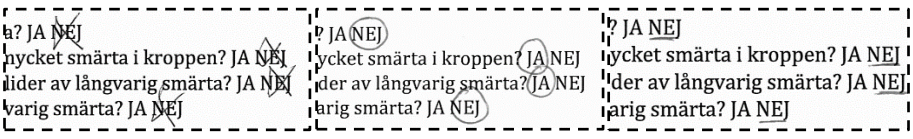


Figure 18. Varying types of binary answer marks in the first version of the main questionnaire.

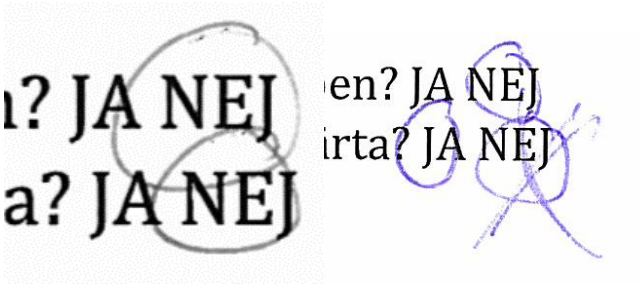


Figure 19. Left: Marks overlapping both JA (YES), NEJ (NO) and another question's answer area. Right: Both JA (YES) and NEJ (NO) are marked.

Two alternatives were considered to handle the encountered problems described above. The first one being improving the computer vision method, the second one being altering the questionnaire to make it more compliant with machine reading.

Thus, a new design of the binary questions that is more in line with previous work within OMR and mark recognition was proposed to the researchers. Using checkboxes, which is common practice in OMR, eased the task of decoding the binary answers, see Figure 20 and Figure 21.

Smärta just nu

- Har du smärta någonstans i kroppen just nu? ☐JA ☐NEJ
  Om JA, var? Vänligen sätt ett tydligt kryss och välj intensitet:

Figure 20. One of the five binary questions in the second version of the main questionnaire.

Långvarig smärta

Långvarig smärta är den typ av smärta som inte går att förklara med tillfällig huvudvärk, träningsvärk, mensvärk eller liknande. Långvarig smärta varar i mer än tre månader i sträck.

- Lider du just nu av långvarig smärta? ☐JA ☐NEJ
- Har du tidigare haft perioder med mycket smärta i kroppen? ☐JA ☐NEJ
- Har du någon i din nära familj som lider av långvarig smärta? ☐JA ☐NEJ
- Är du orolig för att drabbas av långvarig smärta? ☐JA ☐NEJ

Figure 21. Four of the five binary questions in the second version of the main questionnaire.

The two possible answers JA (Eng. YES) and NEJ (Eng. NO) are mutually exclusive options. For example, a person either experiences pain or not, it does not experience pain at the same time as it does not experience pain. The answers are at least intended to be mutually exclusive, however in practice there exists cases when both answers are marked,

see Figure 19. An answer is considered marked if its checkbox is sufficiently enough *more strongly* marked than the alternative answer's checkbox. If none of the checkboxes is more marked than the other, i.e if both are empty or about as much marked, the question is considered unanswered.

Decoding the answers from the binary questions is therefore done by computing the ratio between the pixel sums of the two checkboxes and comparing this to a threshold. The first step in the checkbox answer decoding is to detect the checkboxes in the reference document, and this is done by finding contours and approximating polylines with the douglas-peucker algorithm. See Figure 22. The procedure is similar to identifying digit boxes, described in Section 3.1.11 but with other parameters. The checkbox coordinates are used to slice checkbox images from the skew-corrected questionnaires, which are binarized before computing their pixel intensity sums. A checkbox image pair (img_{YES}, img_{NO}) is considered to have been marked YES if:

$$\frac{sum(img_{YES})}{sum(img_{NO})} > thres, \tag{10}$$

and marked NO if

$$\frac{sum(img_{YES})}{sum(img_{NO})} < \frac{1}{thres}. \tag{11}$$

In the case $\frac{1}{thres} \leq \frac{sum(img_{YES})}{sum(img_{NO})} \leq thres$, the question is considered unmhär säger jag binarized men sen säger jag no thresarked or not sufficiently distinctly marked. The above described method was chosen over other alternatives, e.g. pixel counting and the thresholding method described in Section 2.2.7, for several reasons. To begin with, it is a suitable method considering the questions are both binary and mutually exclusive; only one answer is supposed to be marked. Furthermore, in contrast to what is the case in many existing OMR-techniques described in Section 2.2.7, the checkboxes cannot be expected to be filled in completely. There are no such instructions, and the method thus needs to detect vaguer marks, and marks of different shapes.. The requirement on a checkbox to be significantly more marked than its neighboring checkbox prevents small noise in the image from leading to faulty predictions. By not relying on pre-defined absolute pixel sum thresholds, but instead comparing pixel sum ratios between nearby checkboxes, slight changes in background illumination will not affect the prediction. The decision not to use thresholding before making the comparison, is based on the fact that the nearby checkboxes in each pair can be assumed to have similar brightness. As each checkbox images is only compared to its neighboring checkbox, partially erased answers are not uncorrectly predicted as marked *as long as the other alternative is more strongly marked*.

The above described comparisons will not suffice if either of the checkbox images have pixel sum zero. This is avoided by including the edges of the checkboxes in the comparisons. This way, it is ensured none of the pixel sums are zero. The situation where an unmarked box is predicted to be marked because it contains a few pixels deriving from noise but still has a pixel sum proportionally much greater than its unfilled neighboring checkbox is also avoided.

Adjusting the threshold value affects the behaviour of the method. High thresholds require more distinct marks while lower thresholds pick up more faint marks. A too low threshold results in a higher tendency to pick up noise as marks. The empirically derived threshold used as default in *VASReader* is 1.4, meaning a checkbox must have a pixel sum of at least 140% of its neighbouring check box's pixel sum in order to be classified as marked. See Figure 23 for an example.

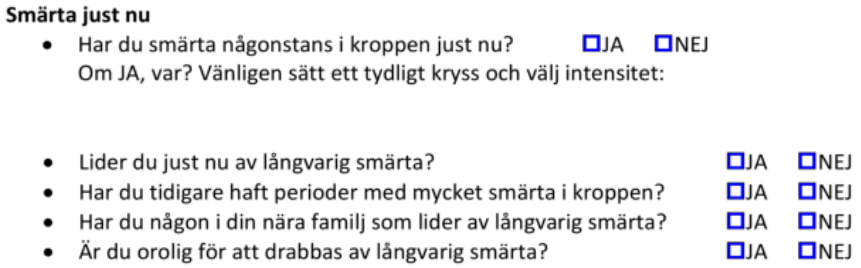


Figure 22. Checkboxes detected in the questionnaire (in blue) using contour approximation.

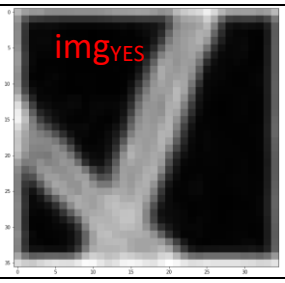
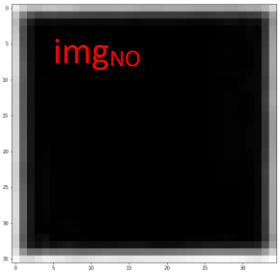
Checkbox answer	Inverted checkbox image	Pixel sum	Prediction
☒JA		$sum(img_{YES}) = 73186$	$\frac{sum(img_{YES})}{sum(img_{NO})} = 2.3674$ $2.3674 > thres = 1.4 \rightarrow prediction = YES$ *according to Eq. 10*
☐NEJ		$sum\ (img_{NO}) = 30912$	

Figure 23. Example of inversion, pixel summation and pixel sum ratio calculation for a checkbox pair.

3.1.10 Associating the Questionnaires with an ID

During the project an unsolved question emerged: how each questionnaire can be associated to the correct participant effectively. The question was both *what kind* of identification to use, and *how* to associate each questionnaire with an id and later associate this id with the participant in the research analysis pipeline. Due to the sensitive nature of the data handled in the PrePain project (e.g. personal number, genetic information), the participants' personal integrity is critical. Identifying the participant by personal information such as social security number is not an option.

A possible solution would be to associate each questionnaire with the unique five-digit MRI number which is retrieved from each unique MRI scanning session. This would not reveal any personal information and could later in the data analysis pipeline be connected to the correct participant by the researchers.

The natural following question was then how this MRI number would be linked to each questionnaire. Alternatives considered were e.g. if the responsible researcher could somehow include this ID in the questionnaire scanning process, e.g. by naming the pdf file, or if the number could be written on the questionnaire and recognized by computer vision. To write the MRI number by hand on the questionnaire was considered the preferred option by the researchers, leading to the least extra-work. The idea was for this number to be read automatically with ICR. Thus, a new task within the framework of the project became to investigate whether handwritten digits could be reliably recognized and

used to identify each questionnaire. The fact that the MRI number would be filled in by the responsible researcher and not the study participant was thought to facilitate the recognition task, considering the researcher would be aware of the fact that the digits would then be machine read. As mentioned by Loke et. al., whether or not the respondent is trained or untrained is one of the factors affecting the difficulty of an OMR-, OCR-, or ICR-task (Loke, Kasmiran and Haron, 2018).

Consequently, a field with five squares intended to contain a digit each was added in the top left corner of the questionnaires' first page, see Figure 24 and Figure 25. The digit squares were complemented with a short instruction to the participant not to fill in the field. In order to distinguish the MRI number field from the parts of the questionnaire which the participants are supposed to fill in, the MRI-number field was given a light gray background. Luckily, handwritten digit recognition is a well-studied subject, and there already exists a publicly accessible large database with handwritten digits, the MNIST database.

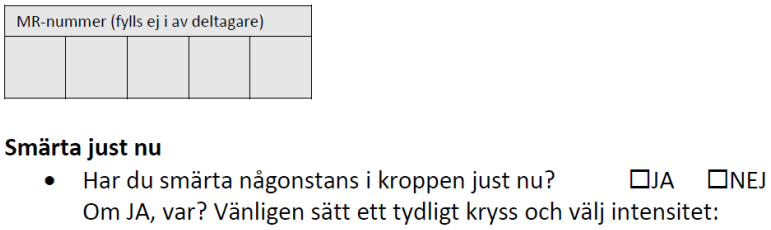


Figure 24. Field for filling in MRI-number in the top-left corner of the questionnaire's first page

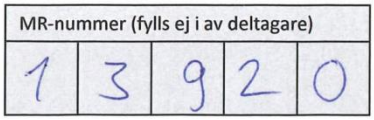


Figure 25. Example of a filled-in MRI number in a scanned questionnaire

3.1.11 MRI Number Recognition

The method for recognizing the MRI-number we propose is to train a convolutional neural network on the images in the MNIST database and use it to predict each digit in the MRI number.

The MNIST database, (Modified National Institute of Standards and Technology database) is a widely used large database containing handwritten digits. The MNIST database is freely available and has a training set of 60,000 images and a test set of 10,000 images (Lecun *et al.*, 1998). It is commonly used to assess the relative performance achieved by different machine learning algorithms and preprocessing techniques. (Deng, 2012)

Each MNIST example is a 28×28 grayscale image in which the digit is size-normalized to fit within an inner 20×20 pixel box. The inner 20×20 pixel box containing the digit is centered in the larger 28×28 image by its pixel mass center (Lecun *et al.*, 1998).

As previously described, the images in the MNIST database have been pre-processed to a very specific format, and for a MNIST-trained CNN to be able to predict new data successfully, the new data should conform to the MNIST image format. One could of course think of alternatives to this approach, such as augmenting training data to an extent so that the model generalizes better to different data formats. As the pre-processing of MNIST images is well-described, the first approach was selected, i.e. "MNISTifying" the MRI digit box images before predicting the digit in the image with a MNIST-trained CNN.

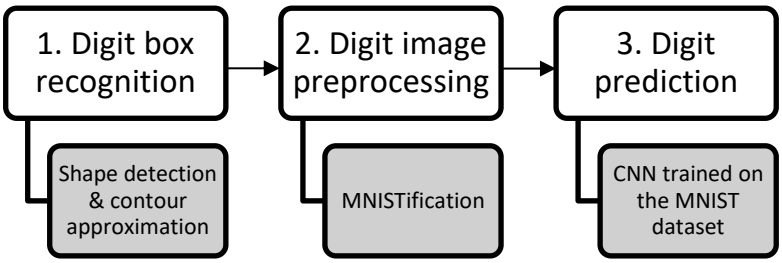


Figure 26. MRI number recognition workflow

The first step in recognizing the MRI number is to detect the MRI digit boxes in the reference questionnaire and extract the coordinates of these boxes, sorted from left to right. This is done through contour approximation, described in Section 2.2.8, implemented with the OpenCV functions `findContours()` and `approxPolyDP()` on the binarized grayscale reference image. To avoid detecting the binary question checkboxes or any other unwanted shapes, a minimum and maximum area is set for the contours retrieved from `findCountors()`, see Figure 27. A list of digit box coordinates $(y\_start, y\_end, x\_start, x\_end)$ sorted from left to right is returned.

Figure 27. Digit boxes detected (in blue) in reference questionnaire

The digit box coordinates are used to slice five digit box images from each skew-corrected questionnaire. Each grayscale digit box image is preprocessed according to a six-step procedure. The preprocessing procedure is constructed based on the description of MNIST images provided by Yann LeCun, Corinna Cortes and Chris Burges (*MNIST*

handwritten digit database, Yann LeCun, Corinna Cortes and Chris Burges, no date). As the goal is to make the image conform to MNIST format, the process can be called "MNISTification". Figure 28 illustrates the output of each step with an example image. Each digit box image is:

1. Inverted and binarized
2. Cropped slightly to remove possible edges remaining from rectangle sides
3. Cropped to only contain digit
4. Resized so digit fits in inner 20×20 box while aspect ratio is kept.
5. Padded with black pixels
6. Shifted so that center of mass is centered in the outer 28×28 box.

In step 1 the image is inverted and binarized using simple thresholding. As can be seen in Figure 28 a) and b), some pixels from the digit box's sides remain in the sliced image. To remove these, the image sides are slightly cropped (see Figure 28 step 2). The crop ratio is chosen sufficiently small to avoid losing any pixels belonging to the digit. The image is then cropped again, by removing all completely black rows and columns surrounding the digit, see step 3 in Figure 28. Left is only the digit without any padding, and this is resized to fit in a 20×20 square, while keeping the aspect ratio, see step 4 in Figure 28. The digit is then padded with black pixels to obtain an image size of 28×28, see step 5 in Figure 28. Finally, the center of mass of the image is calculated and the image is shifted to center this in the bounding 28×28 image. The shift is implemented as an affine transformation which is a linear transformation followed by a vector addition,

$$T = A \cdot \begin{bmatrix} x \\ y \end{bmatrix} + B, \qquad \text{or } T = M \cdot \begin{bmatrix} x & y & 1 \end{bmatrix}^T, \tag{12}$$

with the affine matrix $M = \begin{bmatrix} 1 & 0 & s_x \\ 0 & 1 & s_y \end{bmatrix}$, where s_x denotes the shift in x and s_y the shift in y. $X = \begin{bmatrix} x \\ y \end{bmatrix}$ is the vector representation of the 2D image to translate.

The shift is barely seen in the specific example in Figure 28, step 6. A more visible shift can be seen in Figure 29, in which the digit's position is shifted downwards.

The label, i.e. digit value, of the MNISTified digit box image is predicted by feeding it to a convolutional neural network.

The network is a sequential combination of convolutional layers, max-pooling layers, and dense layers. The network architecture is illustrated in Figure 76 in Appendix F. The CNN was trained on the previously described MNIST data set, containing 60 000 training images and 10 000 test images. Analysis of the training- and validation loss and accuracy as a function of epochs have been used to select an appropriate number of epochs, that gives high accuracy without overtraining the model. An accuracy of 99.41 is at best obtained by this CNN.

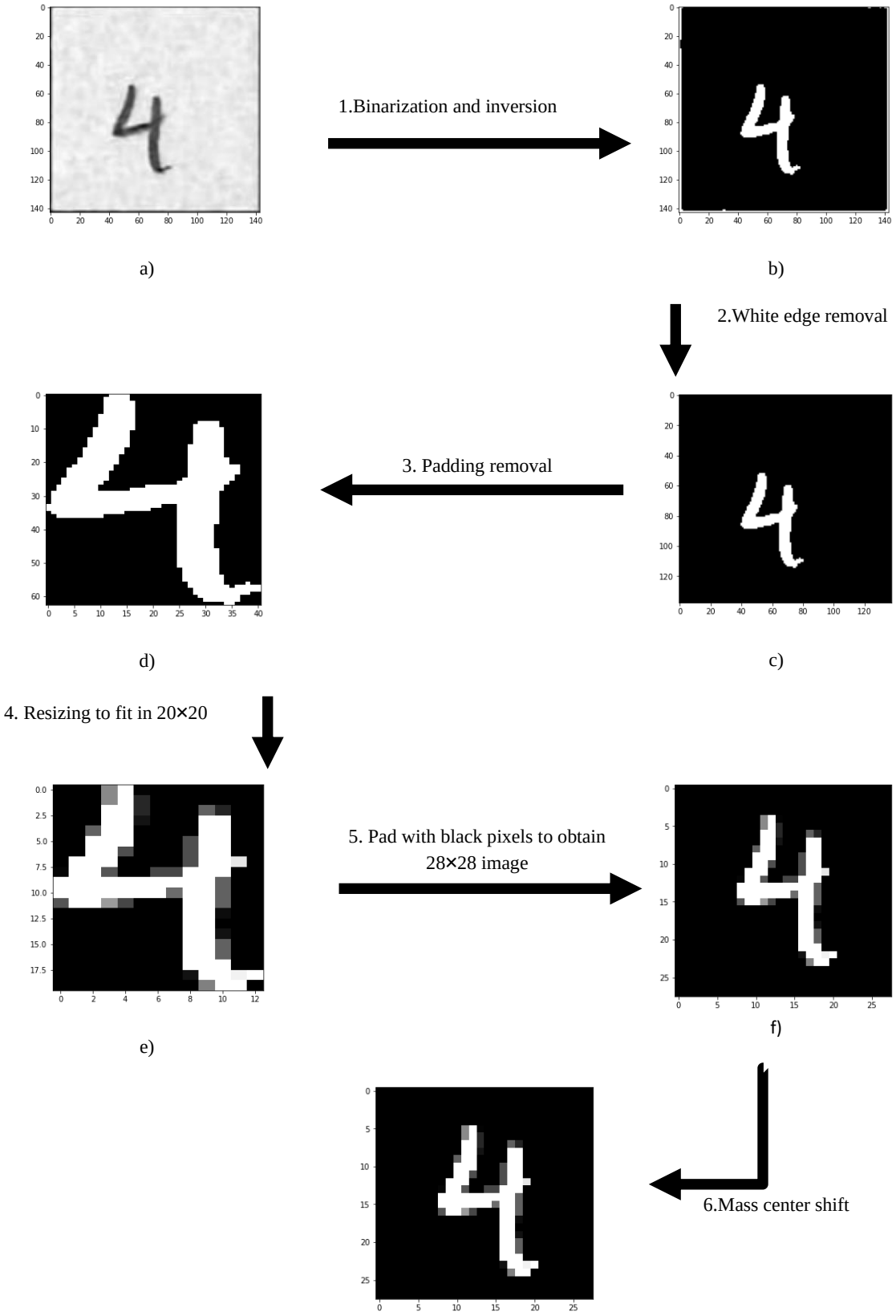


Figure 28. Pre-processing of digit box images

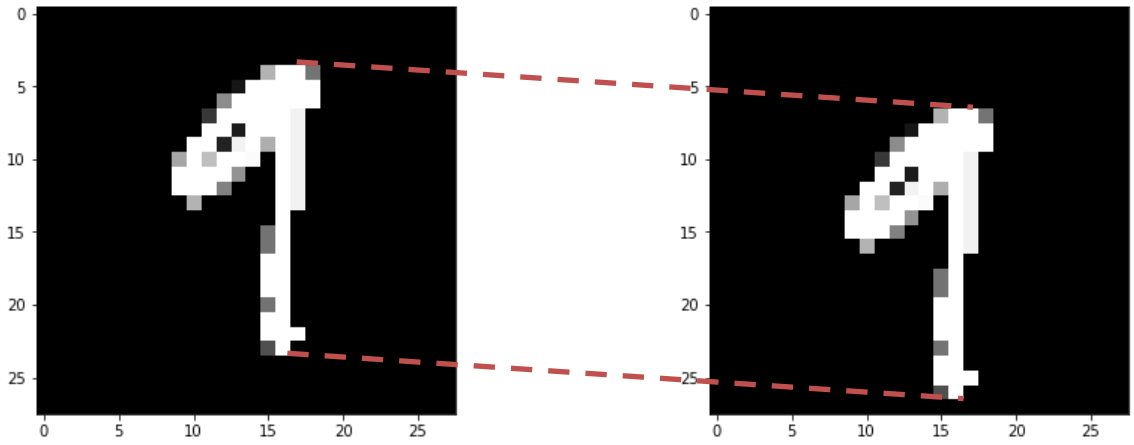


Figure 29. Step 6. The digit's position is shifted downwards so that the center of mass of pixels is positioned in the center of the 28×28 image

3.1.12 Report Generation for Manual Evaluation

To be able to visually validate the questionnaire decoding system's outputted results, a visual report is created for each questionnaire, clearly showing the predictions that the system has made. The cost of faulty measurements and predictions is that incorrect data is saved and used for further analysis. A way of reducing this risk is to demand manual inspection of visual reports before data is saved. Compared to complete manual transcription, this method decreases the researchers' mental workload, as they in most cases only need to accept the output of the computer vision tool. However efficient and flexible the system can be made, there are cases when a computer cannot correctly decode handwritten marks. This is especially the case when a line is marked several times or when marks are crossed over. In some of these cases, a human is able to interpret ambiguous or multiple marks. In other cases, it is difficult for a human too to make a measurement. See Figure 30 for an example of a VAS where it is difficult for a computer and/or human to record marks. Lines with ambiguous marks are flagged. If the VAS mark detection module manages to identify a mark, it is still proposed as a suggestion to the human operator. The visual inspection helps increasing the situation awareness and decrease the risk of erroneous measurements going under the radar.

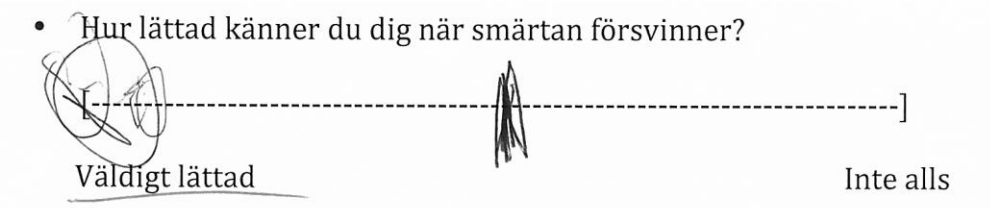


Figure 30. An example of an ambiguously marked line which is difficult both for a human and a computer to interpret. This kind of line will be flagged as ambiguously marked.

3.1.13 Evaluation Metrics

A set of evaluation metrics has been used to evaluate the performance of the developed system. The task of decoding the questionnaires can in turn be decomposed into several smaller and evaluable subtasks. These are depicted in Figure 31 below. The VAS mark flagging, binary question answer marks, and MRI number identifications are classical classification tasks and will be evaluated with conventional metrics, see Table 3 and Figure 32. The VAS measurement is evaluated by comparing the measurements to human-recorded results (using a ruler). For each mark that is recorded by both the system and the human, i.e. each true positive, the difference between the automated computer vision system's output and the manually measured mark is calculated.

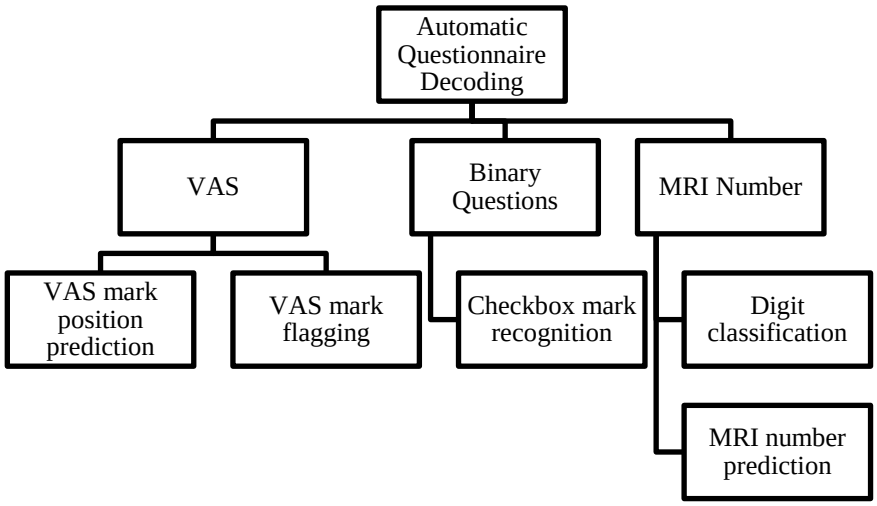


Figure 31. Decomposed evaluable subtasks within the project's first task

		Predicted		
		Positive	**Negative**	
Actual	**Positive**	True Positive (tp)	False Negative (fn)	**Sensitivity** $\frac{tp}{tp + fn}$
	Negative	False Positive (fp)	True Negative (tn)	**Specificity** $\frac{tp}{tp + fn}$
		Precision $\frac{tp}{tp + fp}$	**Negative Predictive Value** $\frac{tn}{tn + fn}$	**Accuracy** $\frac{tp + tn}{tp + fp + tn + fn}$

Figure 32. Confusion Matrix

Table 3. Evaluation Metrics for Classification

Metric	Formula	Interpretation
Accuracy	$\frac{tp + tn}{tp + fp + tn + fn}$	Proportion of all predictions that are correct.
Sensitivity (recall)	$\frac{tp}{tp + fn}$	Proportion of actual positive class correctly predicted.
Specificity	$\frac{tn}{tn + fp}$	Proportion of actual negative class correctly predicted.
Precision	$\frac{tp}{tp + fp}$	Proportion of positive predictions actually positive
F1-score	$\frac{2 * precision * recall}{precision + recall}$	The harmonic mean of the precision and recall.

Table 4. Evaluation Metrics for VAS Measurement Prediction

Metric	Formula	Interpretation
Mean absolute error MAE	$\frac{\sum_{i=1}^{n} \lvert\widehat{y_i} - y_i\rvert}{n}$	Average difference between *VASReader*'s prediction and human's measurement
Root mean squared error RMS	$\sqrt{\frac{\sum_{i=1}^{n} (\widehat{y_i} - y_i)^2}{n}}$	Penalizes large errors
Max absolute error	$\max(\lvert\widehat{y_1} - y_1\rvert, \ldots, \lvert\widehat{y_i} - y_i\rvert)$	The greatest error.

3.2 Results

In this section, the results related to the questionnaire decoding task are presented. In section 3.2.1, the resulting system is described along with a description of the changes made to the questionnaires along the course of the project. This is followed by more detailed descriptions of the results of each addressed subtask in section 3.2.2 -3.2.6.

3.2.1 VASReader

The result of the implementation of above described methods is a computer vision system we call *VASReader*. It is a tool specifically developed to decode VAS questionnaires. In this section, examples from resulting visual reports are shown, which clarify the behavior, capabilities, and weaknesses of VASReader. The evaluation metrics for each individual evaluable subtask are also reported. The results show that it is possible to automatically decode VAS questionnaires– if the design of the questionnaire allows it, and if an appropriate level of automation is selected. Full automation is in this case, and in many other cases, not the appropriate solution when taking into account human performance consequences and automation reliability and costs of decision and action consequences.

The full list of changes in the questionnaires are

- Using solid lines instead of dashed, to ease line detection.
- Using checkboxes for binary questions instead of answer alternatives intended to be encircled.
- Adding an ID field in the top left corner to enable associating the questionnaire to the correct participant.

Albeit the changes in the questionnaire design are small, they facilitate the automatic decoding significantly. See Appendix A and B to view changes.

3.2.2 Reports Generated for Manual Evaluation

An example report from *VASReader* is seen in Figure 33. *VASReader* results are painted or written on the image in colors.

To show which mark detection technique has been used, the three methods described in Section 3.1.5, 3.1.6 and 3.1.7 are color coded in the report. VAS marks detected the first method – *x_detect*, are shown in red, marks detected by the second method – *bracket_detect*, are shown in green, and marks detected by the third method – *mark_detect* are shown in blue. Identified lines are colored yellow. See Figure 34. Each mark is flagged as OK, NO_MARK or AMBIGUOUS_MARK. Ambiguously marked flag can still include a "best guess", if an x-shaped mark or bracket mark is detected. See Figure 35. Unmarked lines are flagged with NO_MARK, see Figure 36. Binary question answers are shown in text next to the checkboxes, see Figure 37. Predicted MRI digits are printed below the MRI number field, see Figure 38.

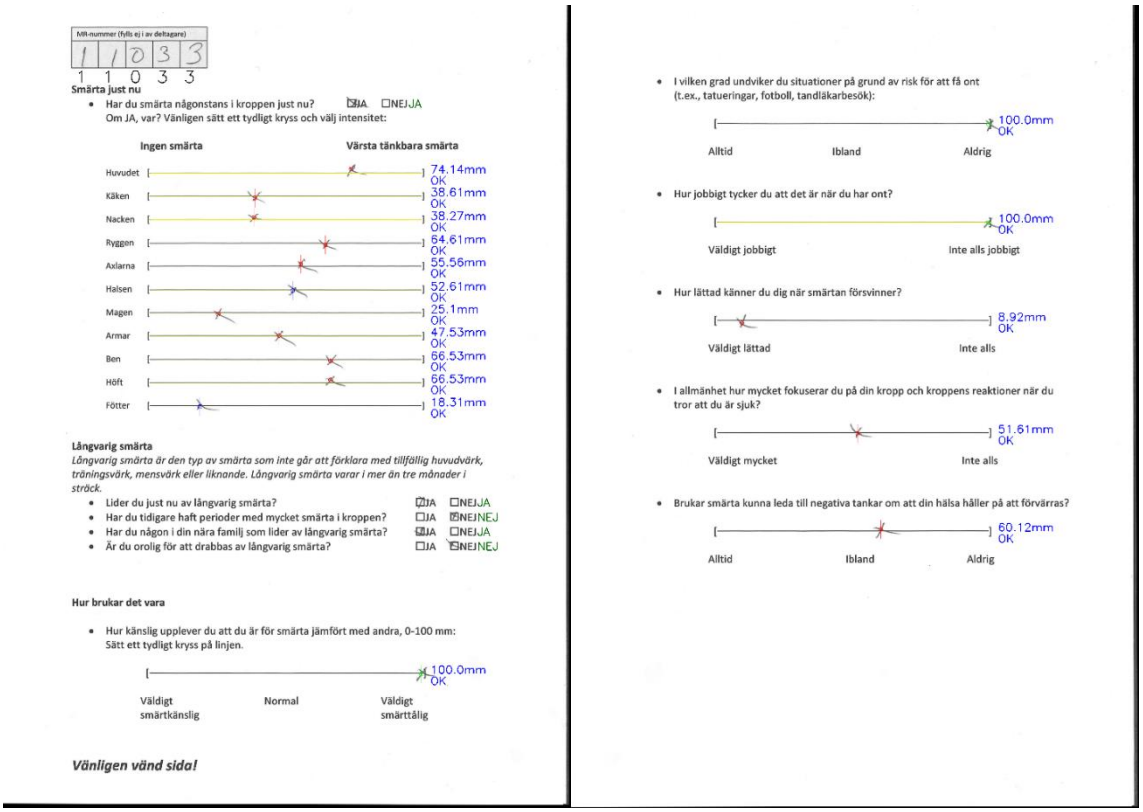


Figure 33. Example of a report from VASReader

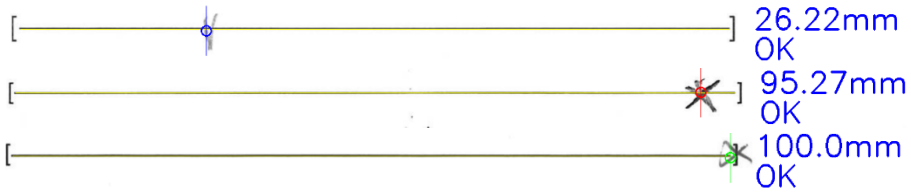


Figure 34. Marks detected by "mark_detect" in blue, by "x_detect" in red, and by "bracket_detect" in green.

Figure 35. Example of a line that is flagged as ambiguously marked. As "x_detect" still detected an x-shaped mark on the line, it is given as a suggestion

Figure 36. Example of a line flagged as unmarked.

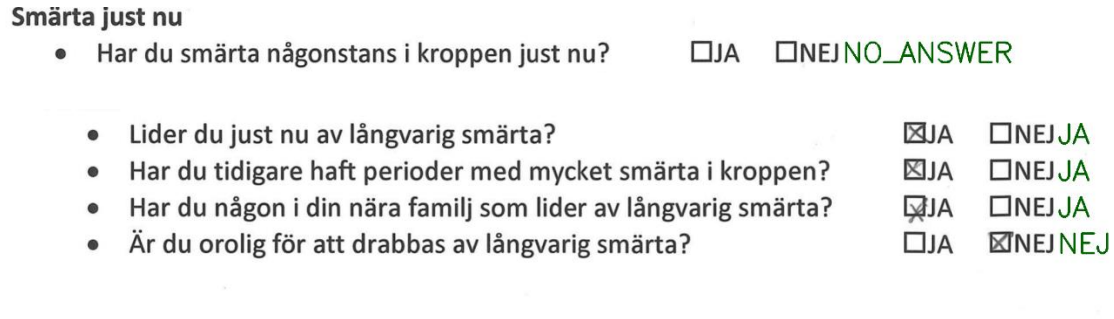


Figure 37. Recognized binary question answers are printed in green next to the checkboxes.

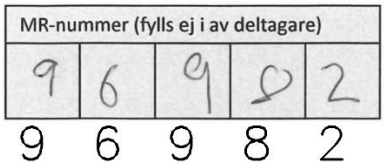


Figure 38. Predicted MRI digits are printed below the MRI number field.

3.2.3 Skew Correction

In Figure 39, the result of the implemented skew correction method is illustrated. The left image has a visible skew angle deriving from misalignment in the scanning procedure. Note e.g. how the lines are not horizontal. The right image shows the image after the homography matrix transformation described in Section 3.1.2. The result is a skew corrected image in which all shapes are aligned with the reference questionnaire (found in Appendix B.1). The full generated visual report for this specific questionnaire is shown in Figure 40, and as can be seen, all measurements and predictions are accurate despite the apparent skew angle in the input image.

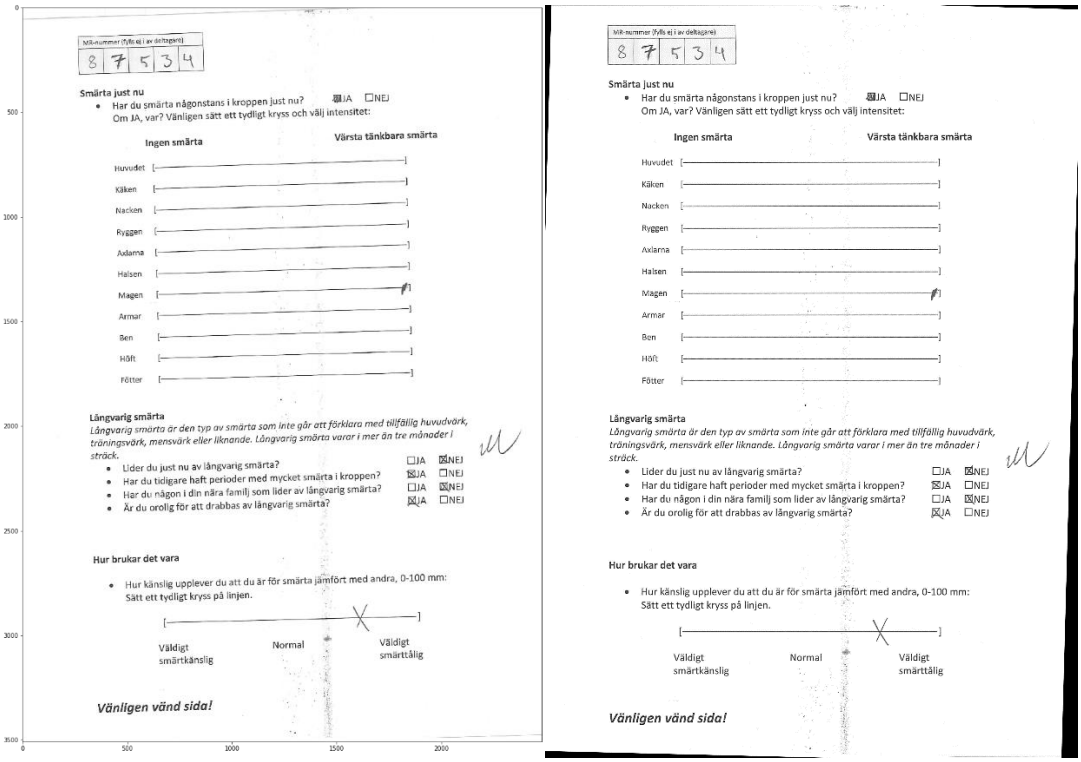


Figure 39. Result of skew angle correction from feature matching and homography matrix transformation technique. Left: original input image with an apparent skew angle. Right: image after skew correction. Note specifically the angle of the VAS lines.

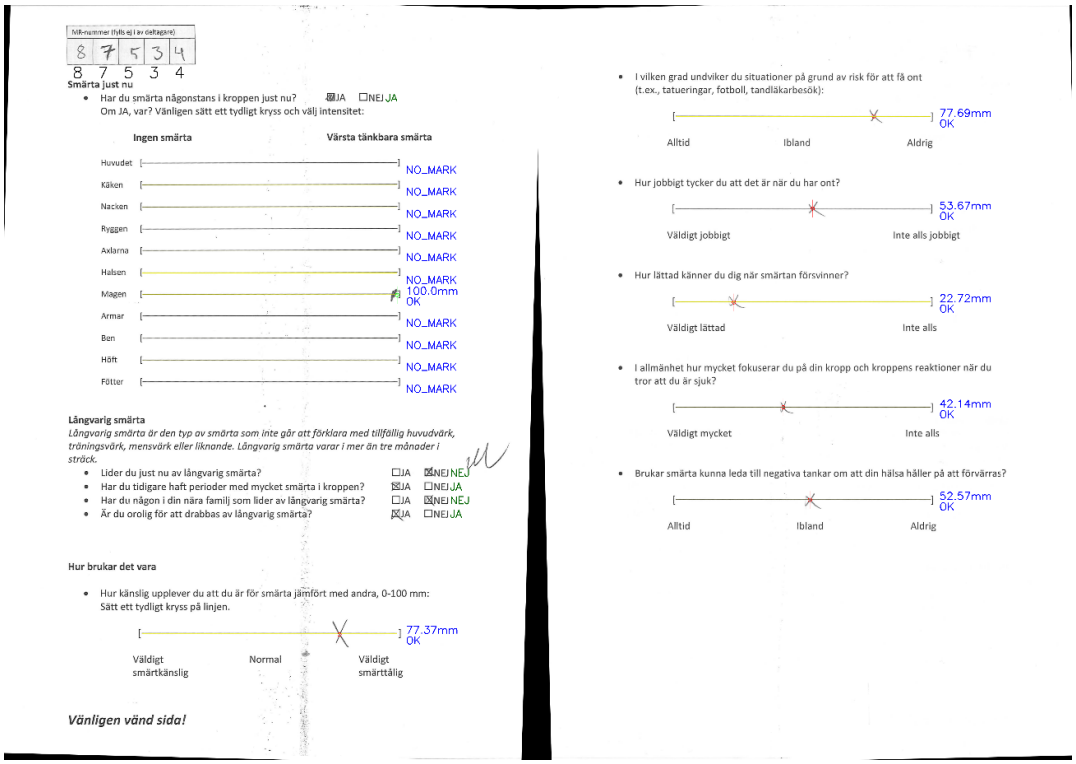


Figure 40. Visual report of questionnaire in which a significant skew angle was present in the input image. Due to successful skew correction, all elements are correctly identified and "VASReader" predictions are accurate.

3.2.4 VAS Mark Detection and Prediction

VASReader detects mark with an accuracy of 98%, when compared to manual measurements. As can be seen in the confusion matrices in Figure 42, and the evaluation metrics in Table 5 and 6, *VASReader* is able detect and measure 661 out of the 672 marks with a mean absolute error of 0.30 mm. The maximum absolute error is 2.90 mm, and that specific case can be seen in Figure 41 below. The difference can be explained by the fact that *VASReader* has detected the first intersection between the line and the mark, while the human has measured the intersection between the x's two legs.

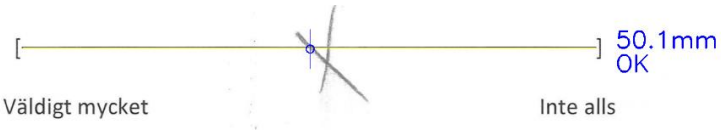


Figure 41. VAS mark with maximum difference between VAS-Reader prediction and human measurement (2.90 mm)

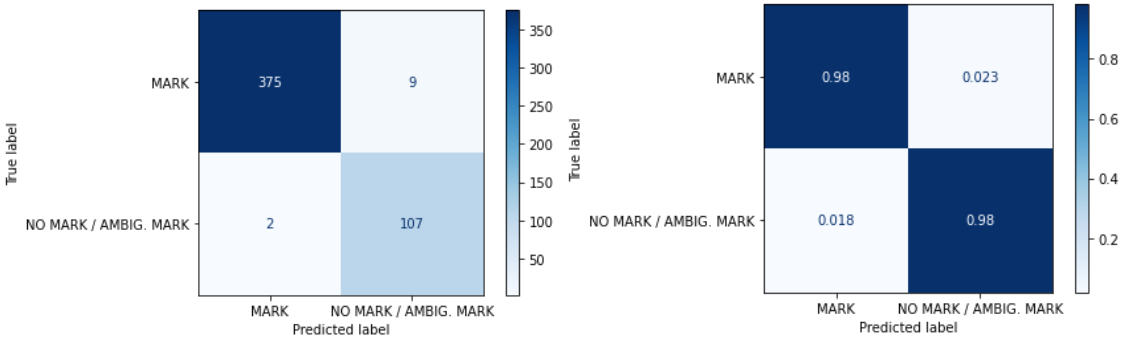


Figure 42. Confusion matrix of VAS mark flags

Table 5. VAS Mark Detection Evaluation Metrics

	Precision	**Recall**	**F1-score**	**support**
MARK	0.99	0.98	0.99	384
NO MARK/ AMBIG. MARK	0.92	0.98	0.95	109
Accuracy			0.98	493

Table 6. VAS Measurement Evaluation Metrics for True Positives

Metric	Result
Mean absolute error MAE	0.30 (mm)
MAE standard deviation	0.41 (mm)
Root mean squared error RMS	1.78 (mm)
Max absolute error	2.90 (mm)

3.2.5 Binary Questions

VASReader predicts binary question answers with an accuracy of 97%, see Table 7. Confusion matrices are found in Figure 43. The F1 scores of labels YES and NO is 0.99 and 0.98 respectively. However the F1 score of NO ANSWER is much lower, only 0.6. This is mainly explained by the class' low support, only four unanswered questions existed in the dataset, and out of these, one was missclassified. A total of 145 questions were included in the test data.

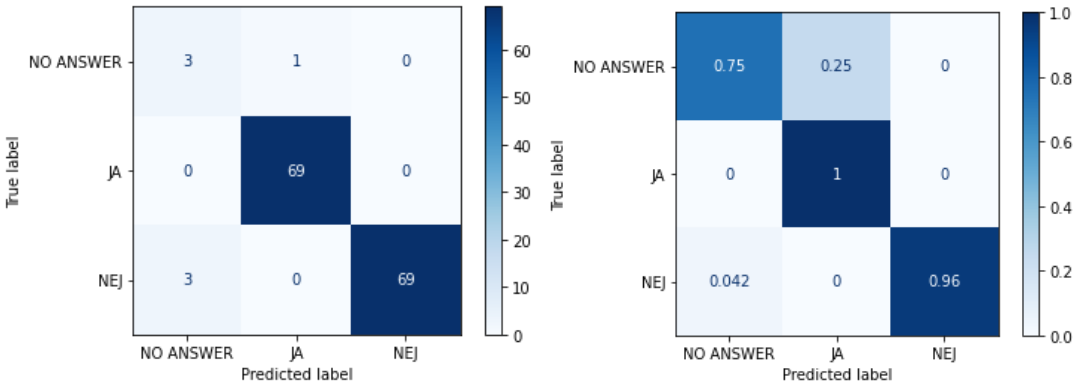


Figure 43. Confusion matrix recognition of binary question answers. Normalized values to the right

Tab1e 7. Evaluation Metrics for classification of binary question answers

	Precision	**Recall**	**F1-score**	**Support**
NO ANSWER	0.5	0.75	0.6	4
YES (JA)	0.99	1.0	0.99	69
NO (NEJ)	1.0	0.96	0.98	72
Accuracy			0.97	145

3.2.6 MRI Number Recognition

The MRI digit prediction an accuracy of 97%. Although this is an ok accuracy, the fact that each MRI number contains five digits, results in an MRI number prediction accuracy of $0.97\text{^}5 = 0.86$. This is not sufficient for reliable automation, and until the performance is increased, it is of extra importance possible misclassifications are corrected when visually inspecting the *VASReader* report. Luckily, this is not a difficult or time-consuming task for the researchers. The test data included a total of 145 digits, and more data would be needed to evaluate the performance. From the confusion matrices in Figure 44, it is evident that nines and twos, threes and fives, and threes and twos are most difficult to distinguish from each other. The precision, recall, F1-score and support for each class (digit) is seen in Table 8.

Inspection of the misclassified digits show two causes of misclassification. To begin with, a portion of the misclassified digits are misclassified simply because their shape resembles the shape of another digit, and the CNN is thus unable to correctly classify the digit, see Figure 45 a) and c). However, a more common source of misclassification is that the digit is not completely contained *within* the digit box. If parts of the handwritten digit are outside the box or on the border of the box, this information is lost when cropping out the digit box image. This is the case in Figure 45 b) and d), and the images fed to the predictor thus lacks the bottom lines of the digits and are hence misclassified.

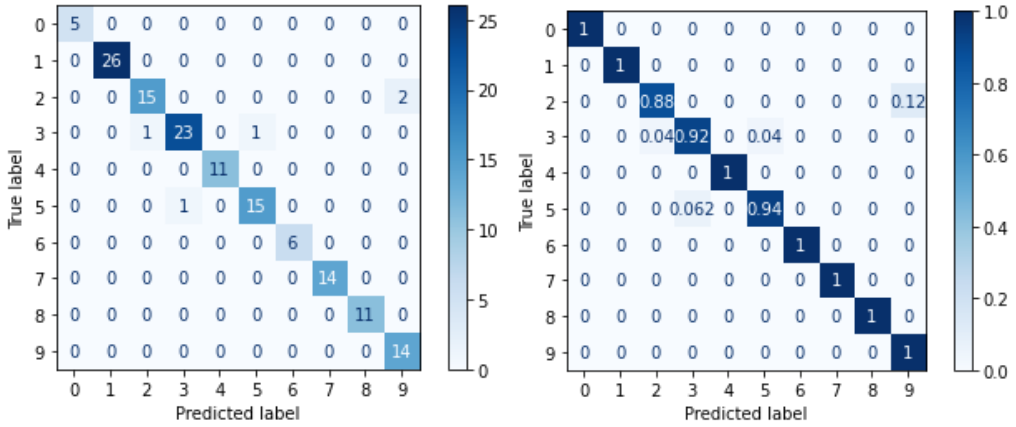


Figure 44. Confusion matrix for handwritten digit recognition with a MNIST-trained CNN. Normalized values to the right

Tab1e 8. Evaluation Metrics for classification of handwritten digits Accuracy highlighted in green

	Precision	**Recall**	**F1-score**	**Support**
0	1.0	1.0	1.0	5
1	1.0	1.0	1.0	26
2	0.94	0.88	0.91	17
3	0.96	0.92	0.94	25
4	1.0	1.0	1.0	11
5	0.94	0.94	0.94	16
6	1.0	1.0	1.0	6
7	1.0	1.0	1.0	14
8	1.0	1.0	1.9	11
9	0.88	1.0	0.93	14
Accuracy			0.97	145

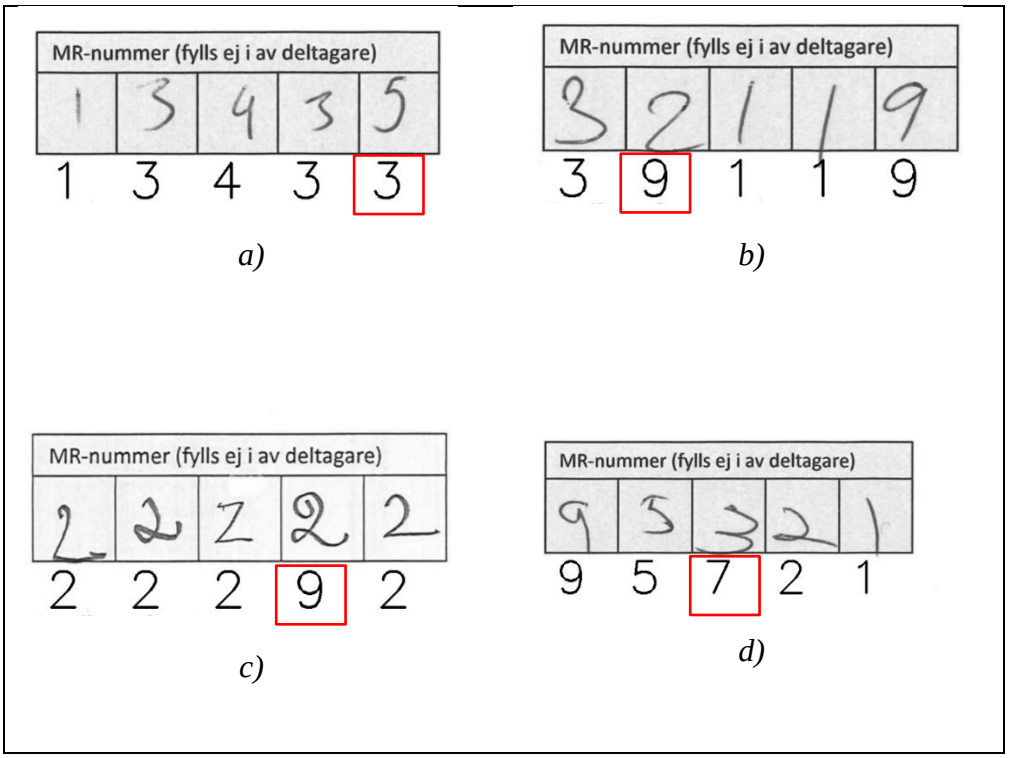


Figure 45. Misclassified digits in red rectangles. a) and c) are misclassified because their shapes resemble the shape of another digit class. b) and d) are misclassified because they are not completely contained within the box, and the bottom parts of the digits are thus not included in the image fed to the CNN.

4. MRIQC Anomaly detection

In this chapter, the methodology and data related to the task of identifying anomalies in MRI data is presented. The MRIQC data is described in section 4.1.1. This is followed by a description of how the anomaly detection technique Isolation Forest is implemented and evaluated in section 4.1.2 - 4.1.4. The obtained results are presented in section 4.2.

4.1 Methodology and Data

In this section, a more thorough description of the MRIQC data is given, followed by a description of the implementation of Isolation Forest on said data. The anomaly detection method chosen to be explored, Isolation Forest, is only one of many outlier detection techniques. This is thus not a thorough comparison of performances of different outlier detection techniques, but rather an attempt to make a small contribution to the work of automating quality assessment of MRI images, with the hope of being able to contribute with some insights about MRIQC data and its outliers. Isolation Forest is however not chosen at random; the methodology selection follows from an explorative data analysis of MRIQC data. An understanding of the dataset, its features and distributions, is the basis of this choice, together with factors such as a favorable time efficiency. One advantage with iForest is that it does not require an assumption of the data being normally distributed, which we shall see is not always the case.

As mentioned in Section 2.3, there are different types of MRI data. T1 and T2 are structural images, BOLD images are functional. Due to this, they have different IQMs. The IQM features for T1 data are presented in Table 13 in Appendix C, and the IQM features for BOLD data are presented in Table 14 in Appendix D. The researchers at the Pain Neuroimaging Lab at KI are interested in applying outlier detection to BOLD and T1 images, as these are the ones that they use in their analysis. It is primarily the functional BOLD images that are important to analyze in this project, as they are utilized for functional brain activity analysis. The T1 images are first and foremost used as a complement to the functional data for the initial planned analyses, e.g. for registration (alignment) of the fMRI data. Although it is possible to use structural data in standalone analyses to study variables such as cortical thickness or folding, such studies are not currently planned within PrePain.

4.1.1 MRIQC Data

A snapshot of the database (‘MRIQC WebAPI - Database snapshot’, 2019) has been used to gain insights about the data and to apply the anomaly detection method. The database snapshot is associated to the article *Crowdsourced MRI quality metrics and expert quality annotations for training of humans and machines* (Esteban *et al.*, 2018). In addition to, for instance, one dataset with curated T1 records and one with curated BOLD records, there is also one dataset containing ratings. This will be discussed further soon. Besides the IQM metric columns shown in Table 13 and 14 in Appendix A and B, there are also

several features containing meta information. As previous research have shown "site-effects", i.e. that IQMs are highly affected by the scanning site they originate from, two of these meta features are regarded as specifically interesting two explore, namely the scanning institution the record comes from, and the machine model that has been used for the scan. Furthermore, the MR Research center at KI uses a 3 Tesla scanner[3], and as different magnetic field strengths could possibly affect the IQMs, another meta feature of interest is the MRI machine's magnetic field strength. The meta features chosen to keep for further analysis are shown in Table 9. The remaining meta features' impact on IQMs and anomaly score is not analyzed in this project.

Table 9. Selected Meta Data Features

Feature	Data type	Interpretation
bids_meta.MagneticFieldStrength	float64	Magnetic field strength of MRI machine (Tesla)
bids_meta.ManufacturersModelName	string	Manufacturer's model name of MRI equipment.
bids_meta.InstitutionName	string	Scanning site

A researcher that runs MRIQC and obtains reports of image data is given the possibility to rate the image quality. If a researcher rates an image, the rating is uploaded to the MRIQC WebAPI along with the anonymized IQMs. Possible ratings are:

1. Exclude
2. Poor
3. Acceptable
4. Excellent

The rating also includes an explanatory comment. Some examples are 'head motion', 'Very low SNR', or 'motion artifacts in dorsal (check coronal view)'.

Although this initially seemed promising for a supervised learning approach, it turned out that most records in the datasets do not have any ratings, and, that there only existed ratings for T1 images, and none for BOLD.

After removing duplicate entries and data points with missing IQM metric values, the cleaned BOLD dataset has 62473 rows, and the cleaned T1 data set has 50883 rows. The cleaned rating data set has 1058 rows. When joining the ratings table with the T1 and BOLD datasets respectively[4] , it became evident that out of the 50883 TI data points, only 430 had ratings. None of the BOLD records were associated to a rating. The distribution

[3] GE 750 Discovery

[4] On the unique identifier *md5sum*

of the rated T1 images is shown in Figure 46. As can be seen, the majority of the images with a rating were rated as “poor” or “exclude”, implying that MRIQC users prioritize to rate low quality images. The distribution of ratings is unlikely to be representative of the distribution of image quality in all T1 data points, considering only 0.85% of all T1 records had ratings. The fact that most datapoints were not associated to a rating lead to the decision of exploring an unsupervised learning technique for anomaly detection.

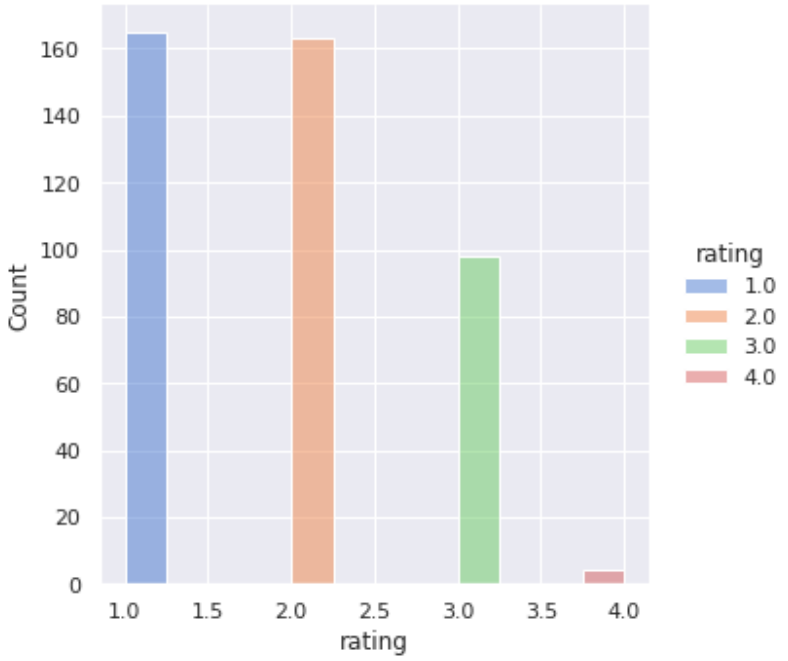


Figure 46. Distribution of ratings in rated T1 data. Only 0.85 of all T1 datapoints are associated to a rating, and out of them, the majority have ratings 1 (=Exclude) or 2 (=Poor)

To get a better understanding of the data and to be able to interpret the results, the MRIQC anomaly detection process starts with an exploratory data analysis (EDA), introduced by Tukey (1977). This is followed by an implementation of isolation forest on TI and BOLD data from the MRIQC database. The trained model is also used to retrieve anomaly scores for a small set of TI and BOLD data from KI of healthy subjects (N=44). The results are visualized and further analyzed through plots, mainly using t-distributed stochastic neighbor embedding (t-SNE). As MRIQC data previously have shown clear site-effects, the relation between anomaly score, scanning site, and machine model is investigated through t-SNE plots. Not all records contain meta information. For this reason, only the subsets of T1 and BOLD datasets containing meta information are used in some visualizations. A subset of the T1 dataset that comes from MRI scanners with magnetic field strength of 3 Tesla is used for training the iForest on T1 data. The reason behind this is that the EDA showed a systematic difference in IQMs from machines of different Tesla values. The distribution of magnetic field strength can be seen in Figure 47 below, and boxplots grouped by magnetic field strengths in Figure 48 show an example of how a feature value differ between these. All datasets that have been used for analysis are seen in Table 10 and Table 11 below.

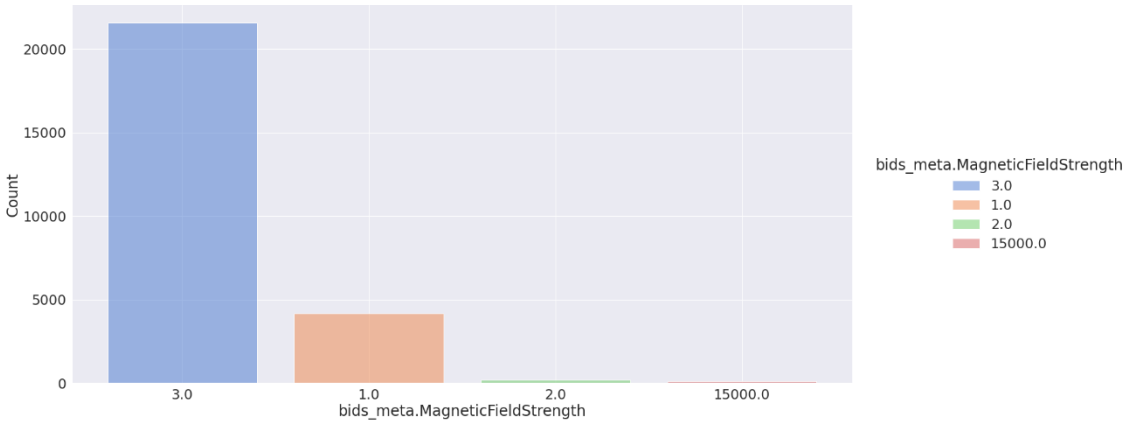


Figure 47. Distribution of Magnetic Field Strength in T1 data

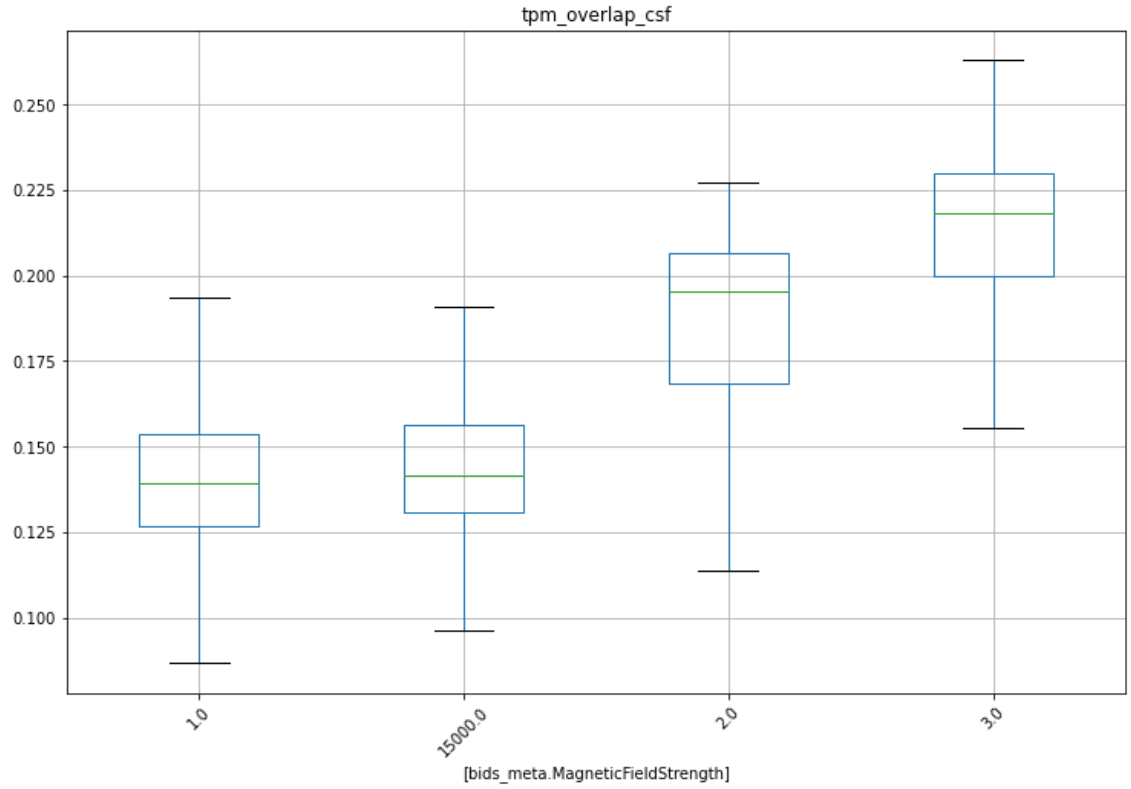


Figure 48. Boxplot of 'tpm_overlap_csf' in T1 data, grouped by magnetic field strength (Tesla) show a systematic difference between machines of different magnetic field strengths.

Table 10. Datasets containing BOLD data

Dataset	Size	Usage
MRIQC Database snapshot of BOLD	62473	EDA + Train Isolation Forest
Subset of MRIQC Database snapshot of BOLD containing meta features	12456	Visualize results and investigate relationship with meta features
BOLD data from KI	44	Test iForest's predictions of KI data

Table 11. Datasets containing T1 data

Dataset	Size	Usage
MRIQC Database snapshot of T1	50883	EDA
Subset of MRIQC Database snapshot of T1 from 3 Tesla machines (containing meta features)	21572	Train Isolation Forest + Visualize results and investigate relationship with meta features
T1 data from KI	44	Test iForest's predictions of KI data

4.1.2 Standard and Altered Implementation of iForest

In addition to the standard implementation of iForest, a slightly altered version of iForest is explored. The IQM features differ from each other in that some are wanted high, others low, and still others should move within a normative range. For example, high values are better for *snr*, which stands for signal-to-noise ratio, low values are better for *qi_1* (AFNIs quality index), while *ICV* (intracranial volume) values should move in a normative range. See Table 13 and 14 in Appendix C and D. Since iForest does not give any information of in which way the point is anomalous, an unusually artifact-free and noise-free record is probable to get a higher anomaly score than a medium quality image, simply because it is anomalously good. One way to handle the problem of not being able to determine whether a data point retrieves a high anomaly score because it is unusually good or unusually bad, is to only let attribute values that lie on the undesirable side of the median of that attribute contribute to the anomaly score. This gives an anomaly score that correlates positively with values that are wanted low, and negatively with values that are wanted high. No tweaking of the scoring is done for attributes that should move within a normative range. The constraint is then that any given directed attribute q only contributes to increase the total averaged anomaly score if it a) is susceptible to isolation; and b) has a value worse than average.

This approach is hereafter referred to as “altered iForest”. Shortcomings of this altered version of iForest are discussed in Section 5.2.

iForest is implemented using the python machine leraning library *scikit-learn,* (short *sklearn*). It should be noted that in sklearn’s implementation of Isolation Forest, anomaly scores are in range [-0.5, 0.5 instead of [0, 1] as proposed by Liu, Ting and Zhou (2008). The translation between the anomaly score s as defined by Liu et al. (described in Section 2.3.2) and the anomaly score as defined by sklearn is simply:

$$\text{anomaly\_score}_{sklearn} = 0.5 - s$$

Sklearn’s version will hereafter be used. Data points with an anomaly score less than 0 are regarded as anomalous. As opposed to the in original definition, anomalous points get lower scores, while normal points get higher scores

4.1.3 Feature Selection

Some of the IQM features for both BOLD and T1 are highly correlated, and likely to be linearly dependent on other variables. See correlation matrix for T1 data in Figure 49, and correlation matrix for BOLD data in Figure 51. Features with high positive or negative correlation are depicted as red and blue areas in the correlation matrix. Not surprisingly, features linked to the same measurement such as *fwhm_x, fwhm_y, fwhm_x,* and *fwhm_avg* are highly correlated. This accounts for the summary statistics as well, for example *summary_bg_mean* and *summary_bg_median* are highly correlated. Correlated features can affect the result of iForest. To begin with, it leads to an unnecessary high number of features, and iForest works best on a subset of the feature space. Secondly, having many correlated features can cause the anomaly score to be more based on these, as they together have a higher probability to be randomly selected as an attribute q in the training phase. See Section 2.3.2 and specifically the definition of iTrees. For this reason, highly correlated features are removed before fitting an Isolation Forest on the data. The resulting correlation matrix for selected features in T1 data is seen in Figure 50, and the resulting correlation matrix for BOLD features is seen in Figure 52. A full list of the selected BOLD and T1 features is found in Table 13 in Appendix E.

4.1.4 Evaluating iForest’s Performance on MRIQC Data

The evaluation of an unsupervised anomaly detection technique is not as straightforward as the evaluation of questionnaire decoding. Evaluating the MRIQC anomaly detection is more difficult, as it is unsupervised, and there exists no “true” answer. For this reason, focus will be on exploring the data through visualization techniques and understanding the behavior of the implemented Isolation Forest. The trained machine learning model is also used to predict a relatively small MRIQC data set from the Pain Neuroimaging Lab at Karolinska Institutet, to see how it performs on KI data.

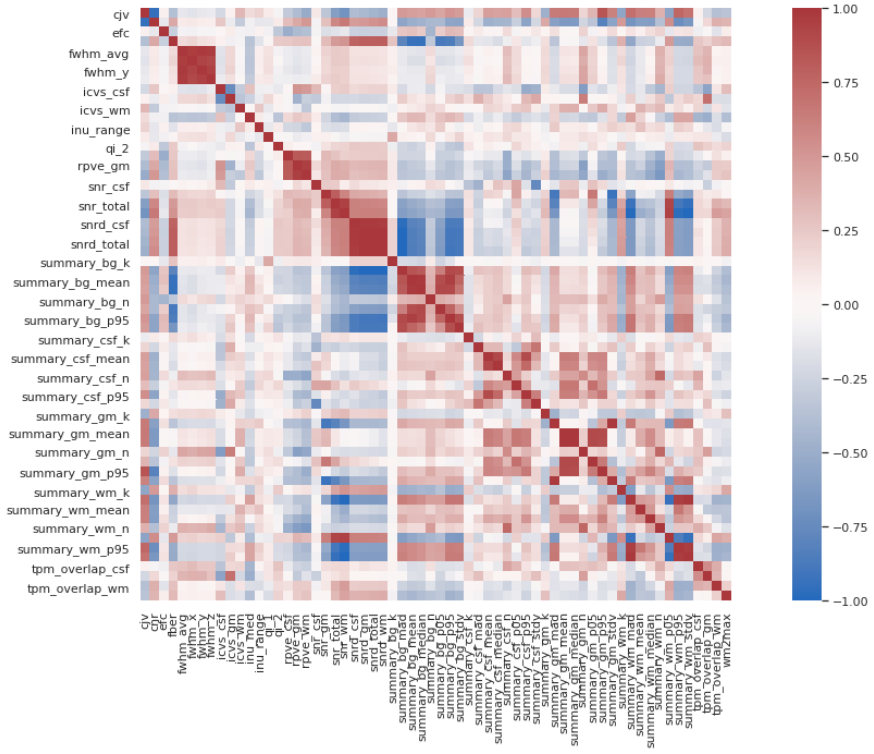


Figure 49. Correlation matrix (Spearman's rank correlation) for IQM features in T1 data.

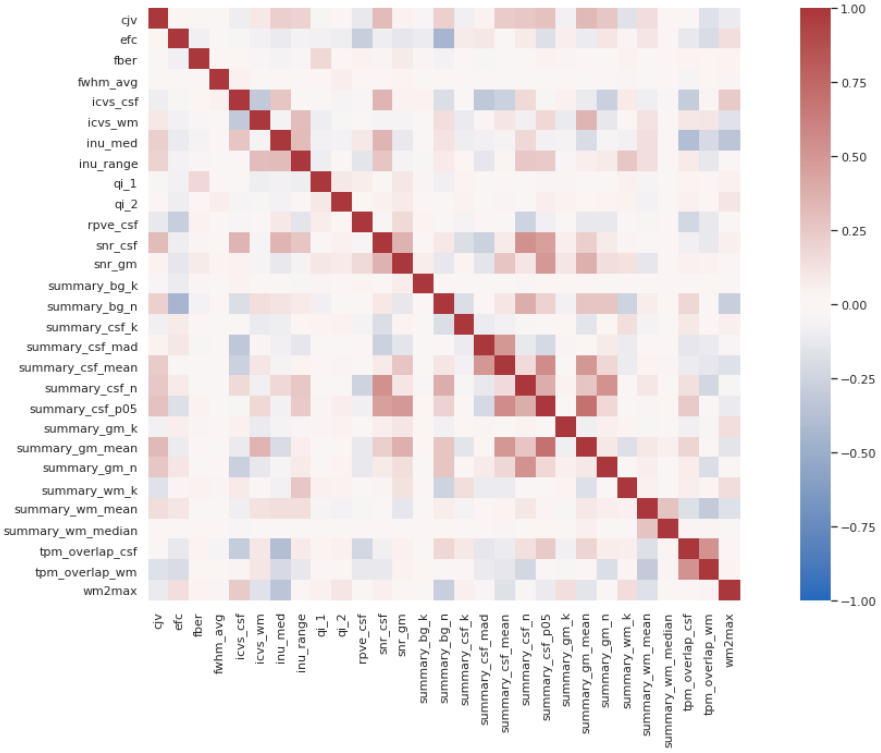


Figure 50. Correlation matrix (Spearman's rank correlation) for IQM features in T1 data after removing highly correlated (ρ>0.7) features

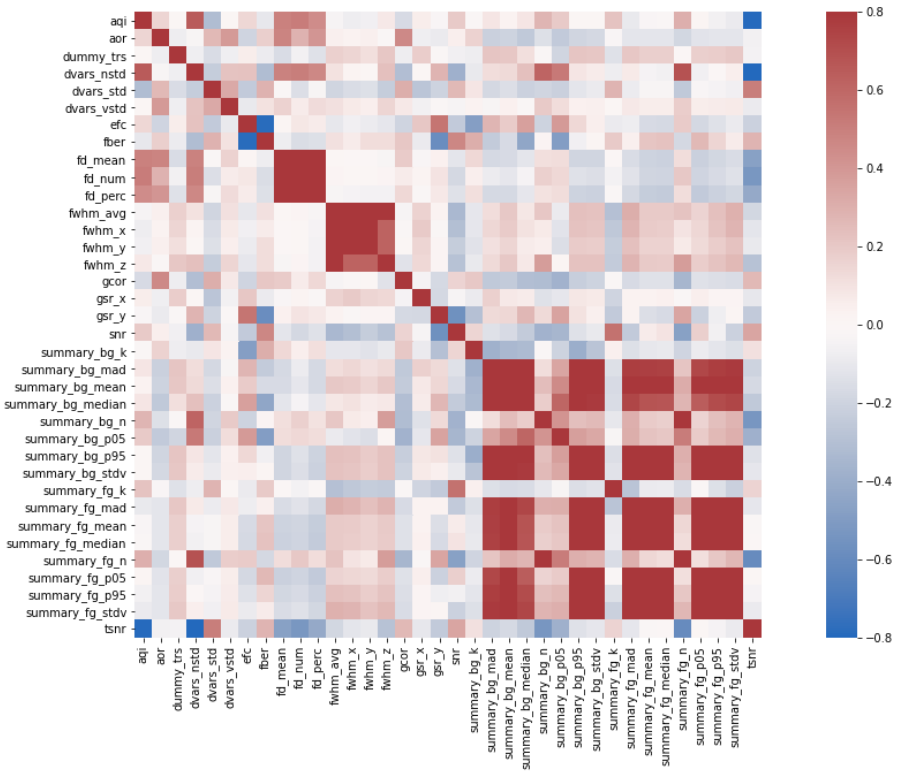


Figure 51. Correlation matrix (Spearman's rank correlation) for IQM features in BOLD data.

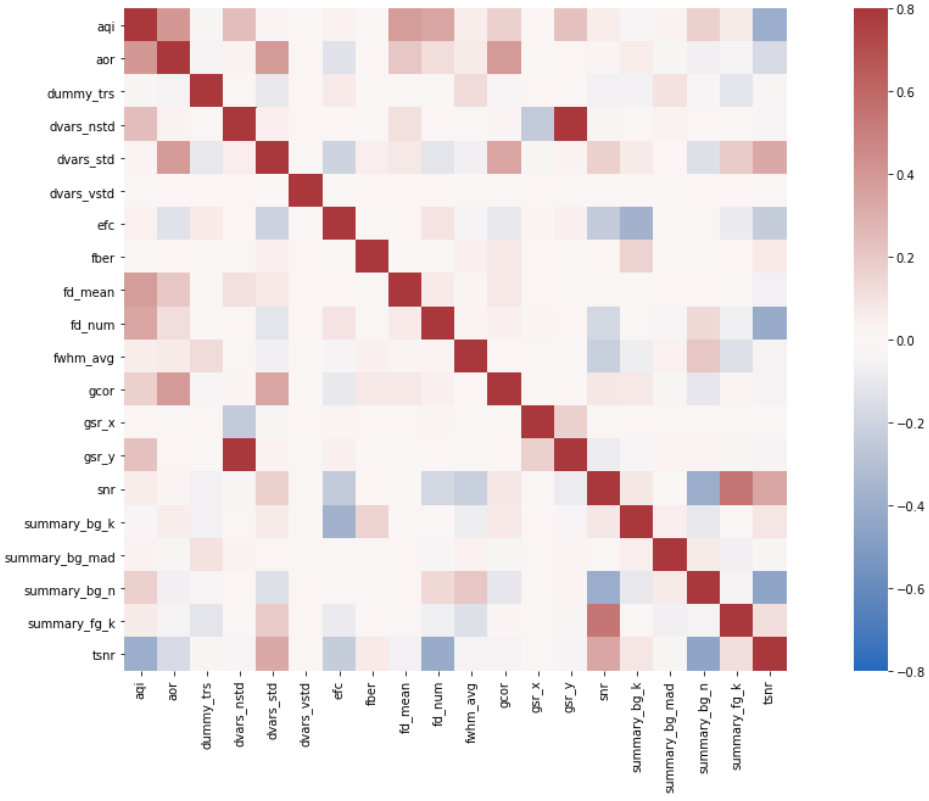


Figure 52. Correlation matrix (Spearman's rank correlation) for IQM features in BOLD data after removing highly correlated (ρ>0.7) features)

4.2 Results

The results of the exploratory data analysis and the anomaly scores obtained using iForest are presented in the following sections.

4.2.1 Exploratory Data Analysis

EDA is an exploratory process meant to aid understanding, interpretation and analysis of data. An initial EDA provides a basis for appropriate selection of data and methods, as well as appropriate interpretation of the results.

Figure 53 shows the distribution of machine models, and as can be seen, most of the BOLD data originate from scans performed in Prisma and Tim Trio MRI scanners. The distribution for T1 data is seen in Figure 54. Notable is that Tim Trio is used in majority of the T1 data. The top five contributing institutions to BOLD data are National Institute on Drug Abuse (NIDA), National Institute of Health (NIH_FMRI), The university of Texas at Austin Imaging Research Center, McGovern institute for Brain Research and The Stanford Center for Cognitive and Neurobiological Imaging (CNI). T1 data comes from an even greater variation of sites. As Esteban *et al.* (2017, 2018) have already shown, the data is structured, or naturally clustered around, its site of origin. It is also dependent on the machine used. Figure 55, which shows the density plot of the BOLD feature *snr* (signal-to-noise-ratio) illustrates this. It is also evident that the attribute values cannot be assumed to be normally distributed, which emphasizes that *iForest* is a suitable method as it does not build on an assumption of normal distribution. Pairwise plots of the first four principal components of T1 and BOLD data respectively, colored by machine model are shown in Figure 56 and Figure 57. Colored clusters indicate that the data is naturally clustered by the machine model and institution. This is further strengthened by the boxplots shown in Figure 58, in which it is possible to see clear inter-machine differences in two exemplifying T1 features. This indicates that there is a possibility anomaly scores retrieved from MRIQC data can be systematically affected by the scanning site and machine. This hypothesis will be further explored in Section 4.2.4.

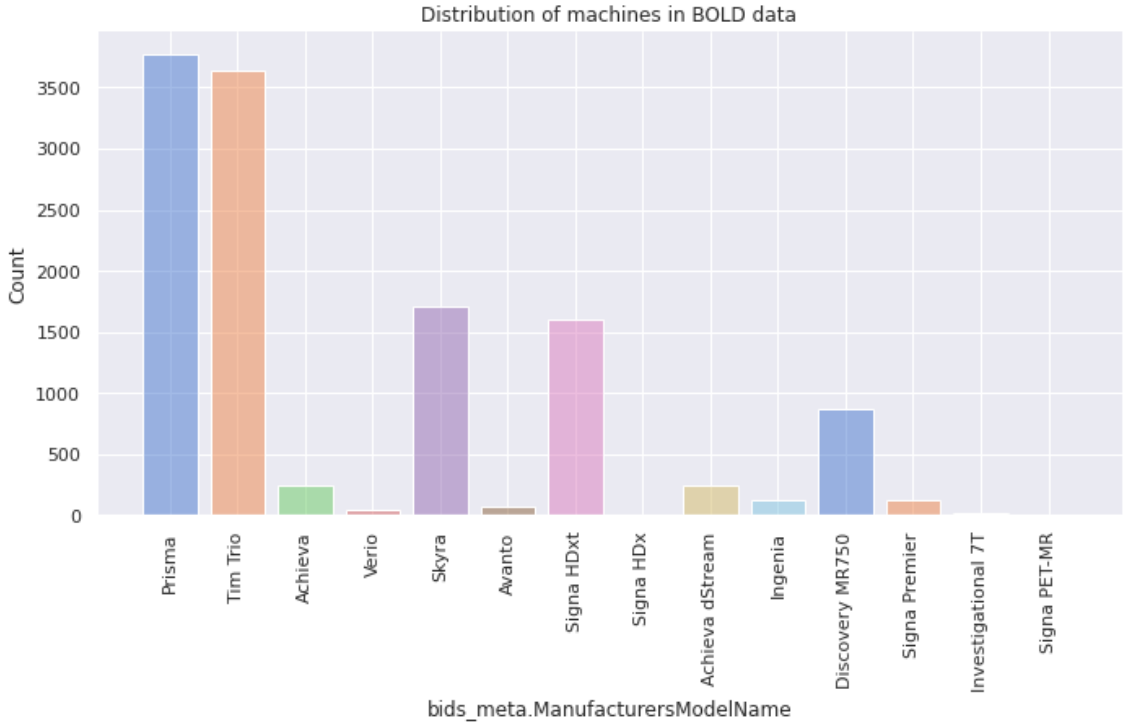


Figure 53. Distribution of machine models in BOLD data.

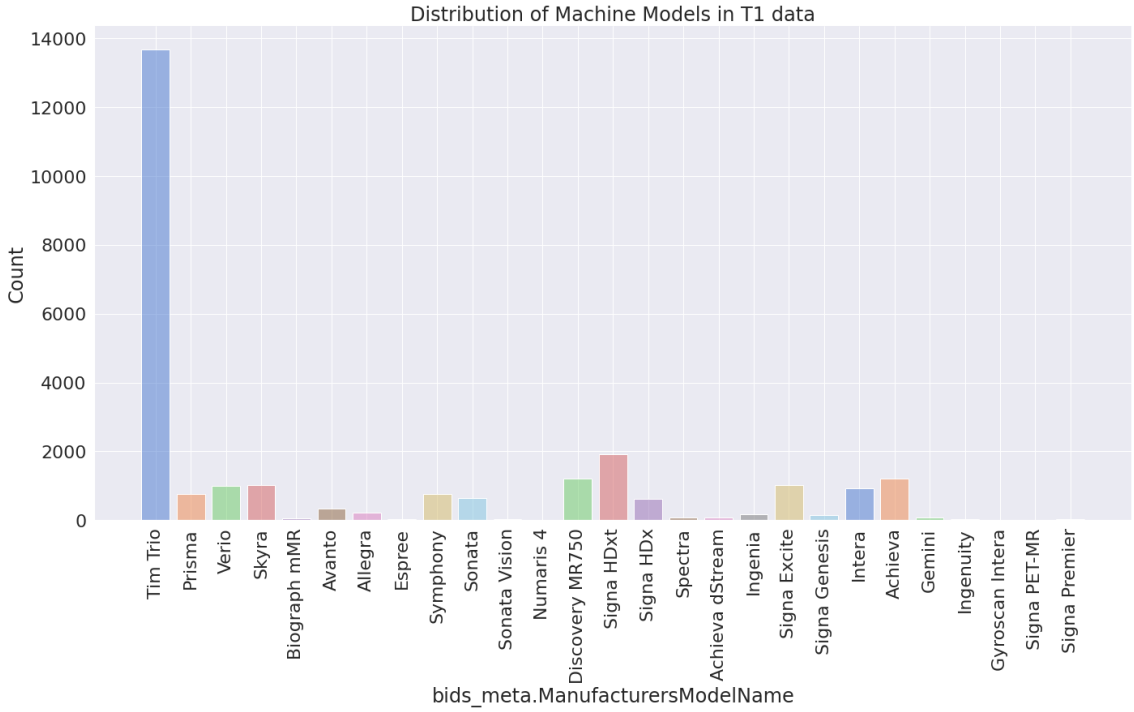


Figure 54. Distribution of machine models in T1 data

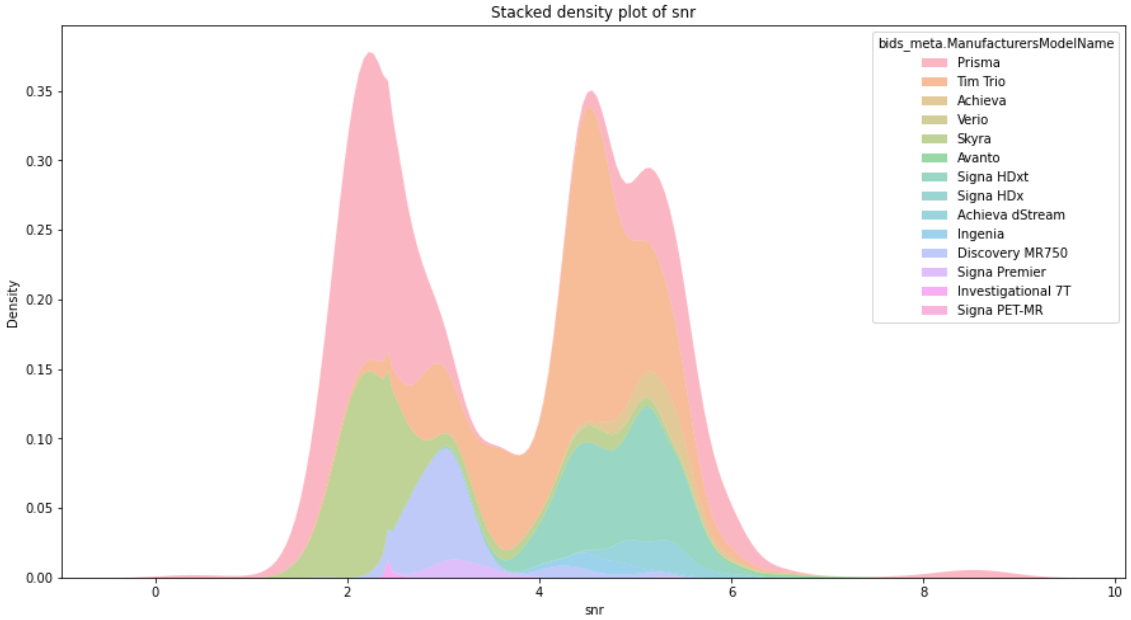


Figure 55. Stacked density plot of feature "snr" (signal-to-noise-ratio) in BOLD data. Colored by Machine model. It is clear that the signal-to-noise-ratio is not normal distributed, and highly affected by machine model

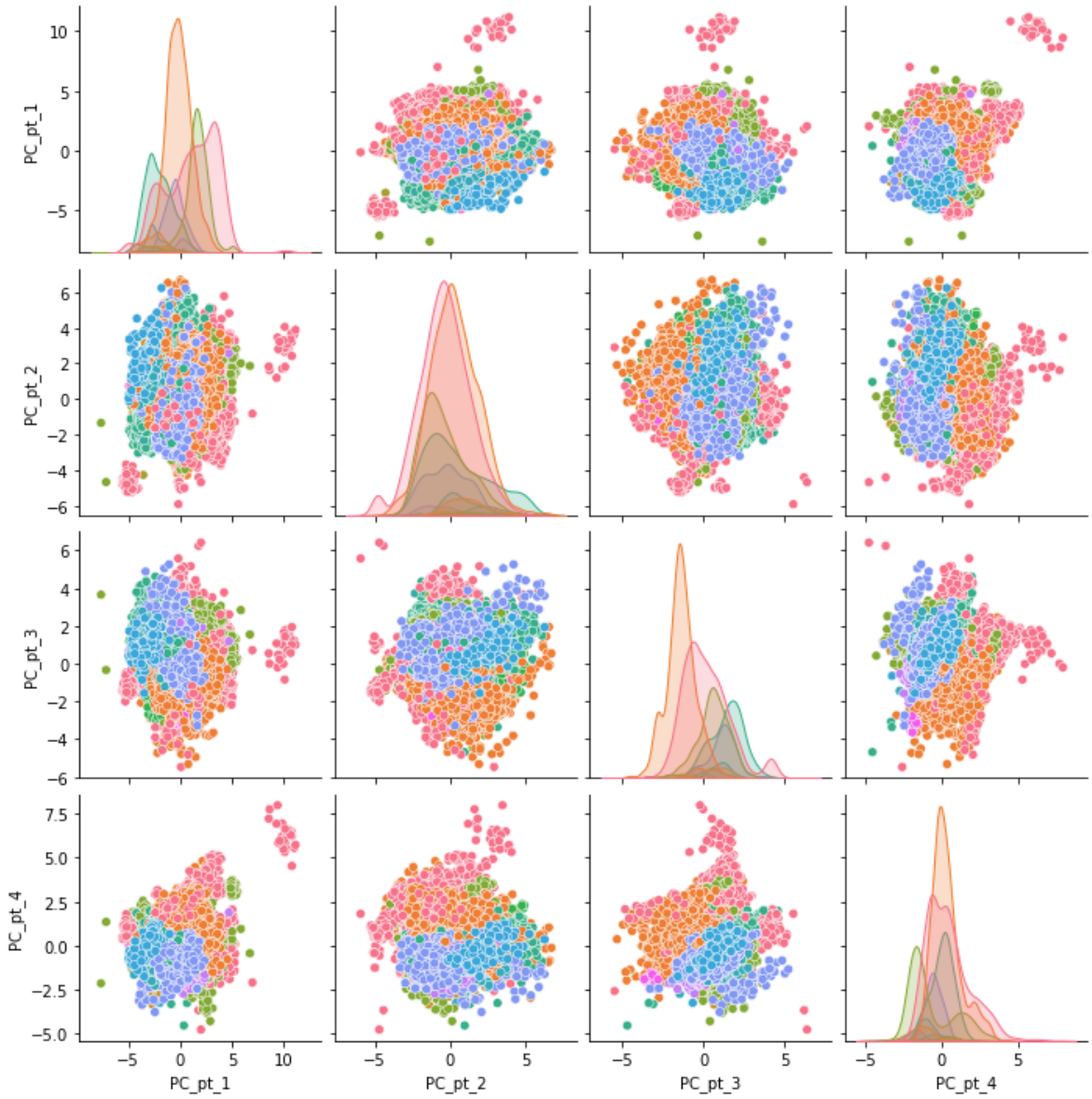


Figure 56. Pairwise plots of 4 principal components in BOLD data (selected features). Colored by Machine Model. Legend is not included due to its size.

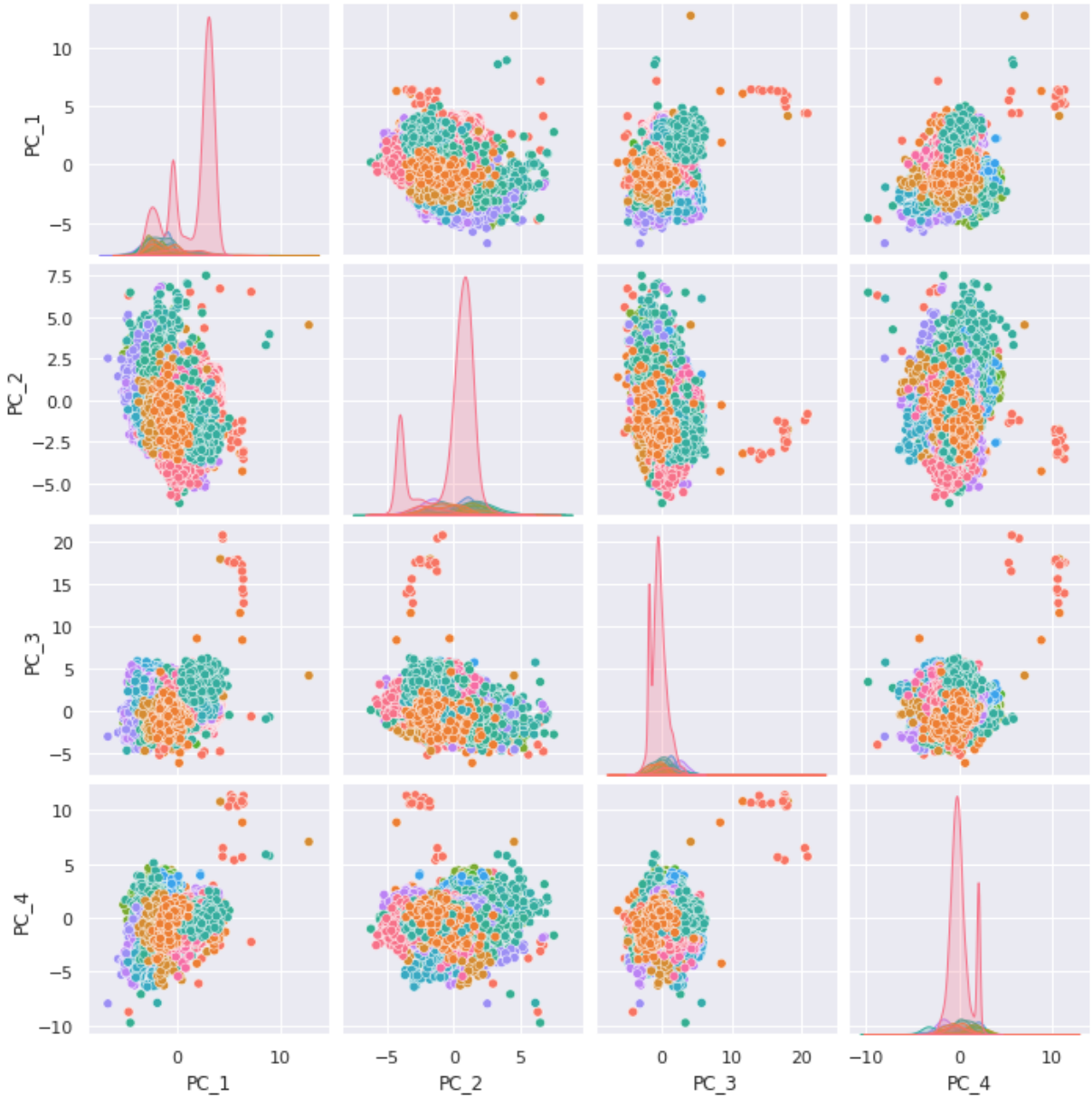


Figure 57. Pairwise plots of 4 principal components in T1 data (selected features) Colored by machine model. Legend is not included due to its size.

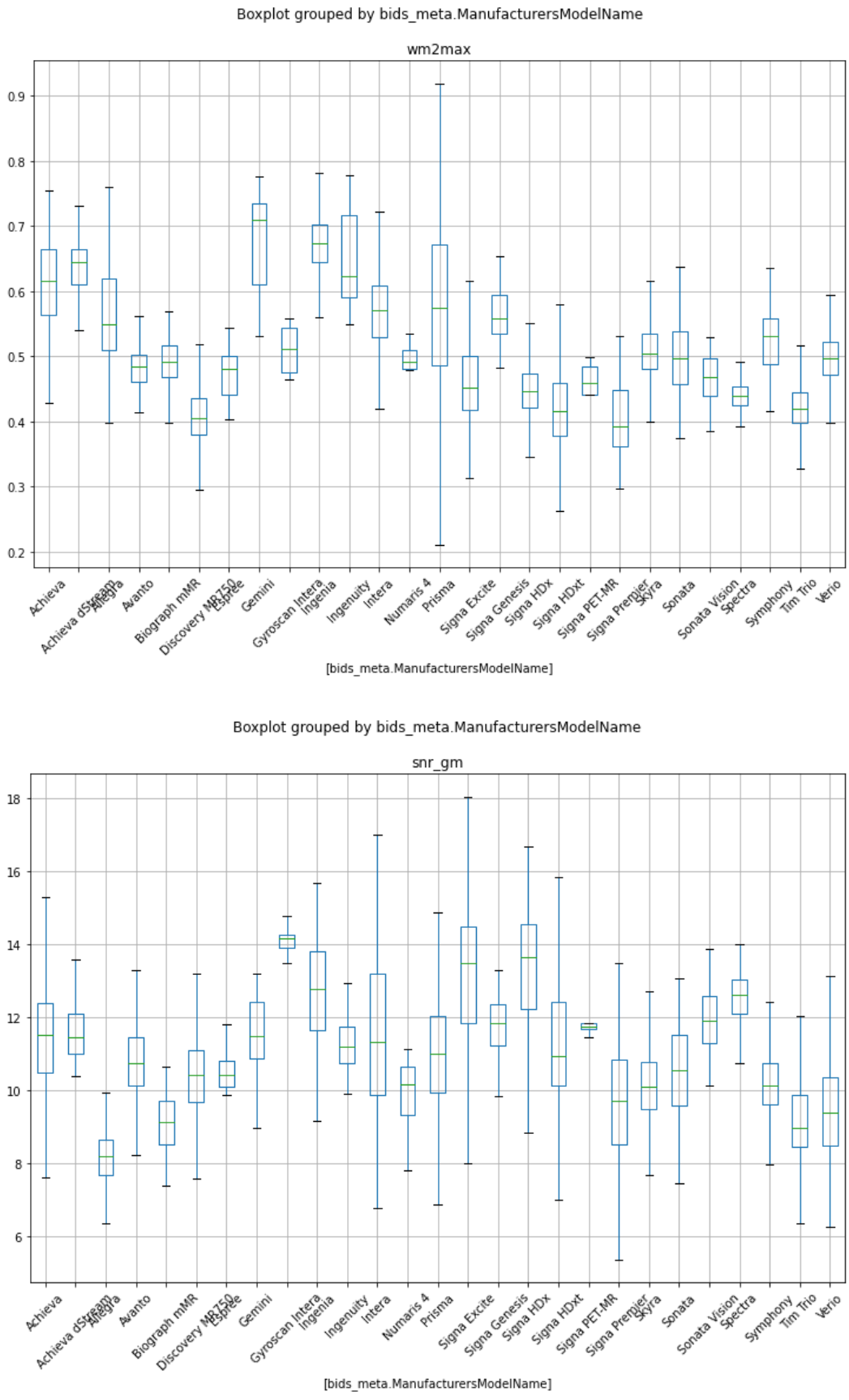


Figure 58. Boxplots of T1 data features "wm2max" (up) and "snr_gm" (below), grouped by machine model

4.2.2 Anomaly Scores Reflect the Underlying Distribution

To understand the behavior of iForest, it is helpful to start off by analyzing not the total averaged anomaly score, but how the scores behave for each feature. The total score is, as described in Section 2.3.2, the average of the scores obtained from fitting a number of iTrees to randomly sampled features.

Although Isolation Forest, unlike some other anomaly detection techniques, is non-parametric (i.e. does not require any assumption of the underlying distribution of the data), the anomaly score reflects the shape of the underlying distribution. Outlier regions correspond to low probability areas. Figure 59 below shows the distribution of *efc* (entropy-focus-criterion) in BOLD data., The blue bins at the tales of the distribution contain datapoints that are classified as anomalous for this specific attribute. The Figure below, Figure 60, show the anomaly score as a function of *efc* values, and it is clear that the score reflects the distribution of the data points seen in the Figure 59. Another example, with the feature *fd_num* (framewise displacement) which is not normally distributed, is seen in Figure 61 and Figure 62. The fact that high anomaly scores reflect low-probability values is clear here too. Entropy-focus-criterion is an example of a directed feature, where values are wanted high. As can be seen in Figure 59, both values less than 0.4 and values higher than 0.6 are scored lower than 0.5, i.e. will contribute to a decrease in the total anomaly score. In the altered version of iForest, only the anomalously bad valued data points are scored high. A comparison between the behavior of the standard implementation of iForest and the altered version of iForest is shown in Figure 63 and Figure 64.

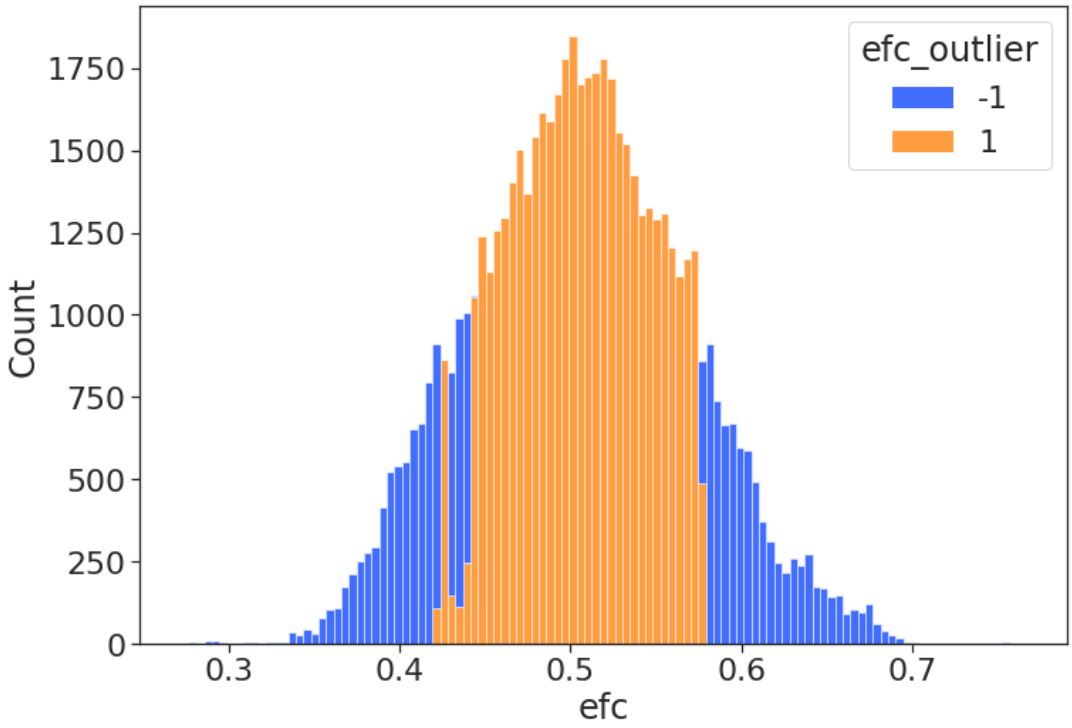


Figure 59. Histogram of attribute "efc" (entropy-focus-criterion) distribution in BOLD data. Blue bins contain datapoints classified as outliers, orange bins contain normal datapoints (Note: The anomaly score this specific attribute contributes with, not the average anomaly score)

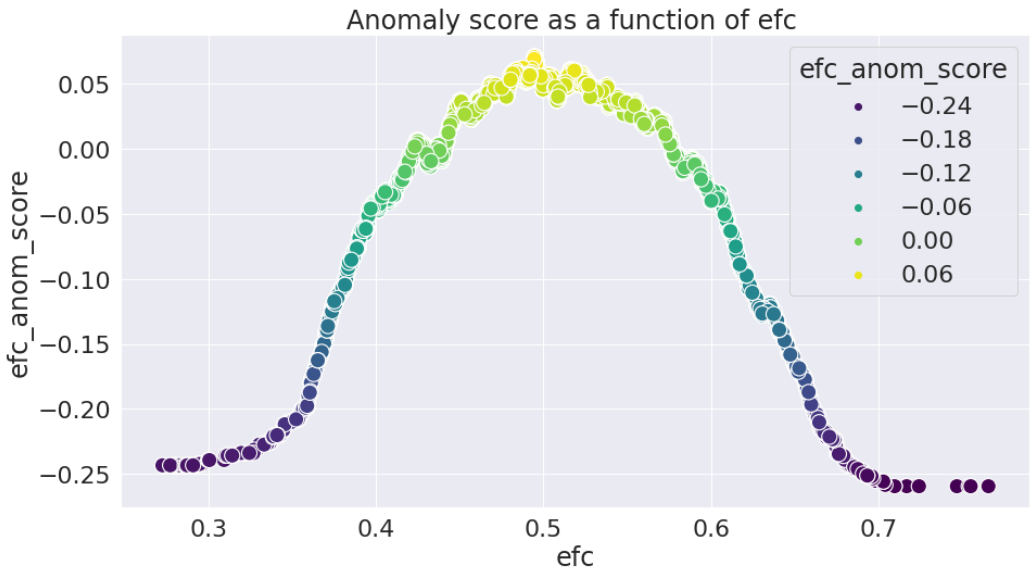


Figure 60. Anomaly score as a function of 'fd_num' (framewise displacement) attribute in BOLD data. The darker the point, the higher the anomaly score. The anomaly score reflects the distribution in the underlying data, see Figure 60 above. (Note: The anomaly score this specific attribute contributes with, not the average anomaly score)

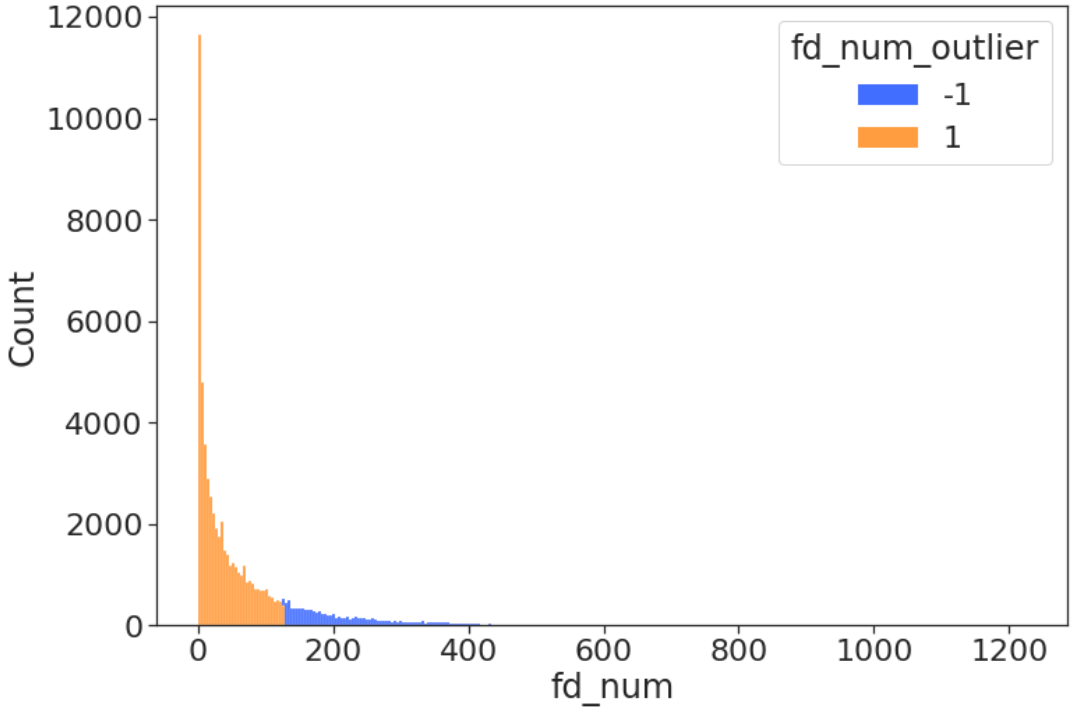


Figure 61. Histogram of attribute "fd_num" (framewise-displacement) distribution in BOLD data. Blue bins contain datapoints classified as outliers, orange bins contain normal datapoints (Note: The anomaly score this specific attribute contributes with, not the average anomaly score)

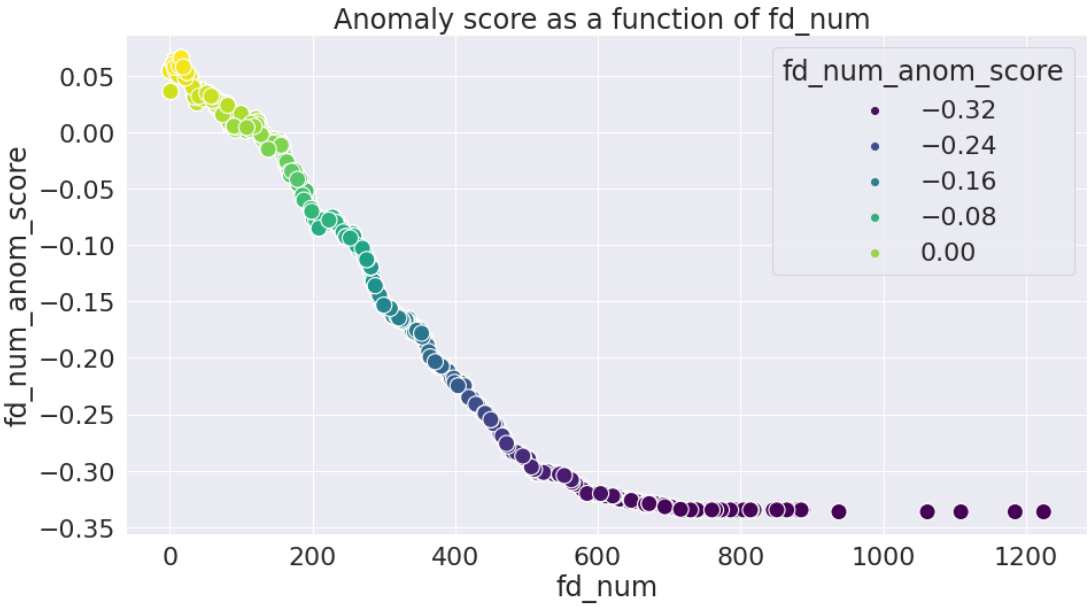


Figure 62. Anomaly score as a function of "fd_num" (framewise displacement) attribute in BOLD data. The darker the point, the higher the anomaly score. The anomaly score reflects the distribution in the underlying data, see Figure 61 above. (Note: The anomaly score this specific attribute contributes with, not the average anomaly score)

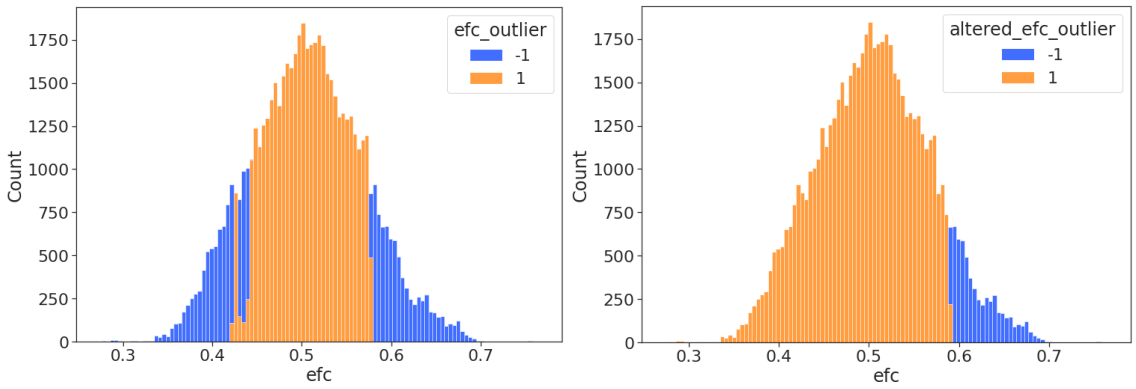


Figure 63. Histogram of attribute "efc" (entropy-focus-criterion) distribution in standard (left) and altered (right) implementation of iForest on BOLD data. Blue bins contain datapoints classified as outliers, orange bins contain normal datapoints (Note: The anomaly score this specific attribute contributes with, not the average anomaly score)

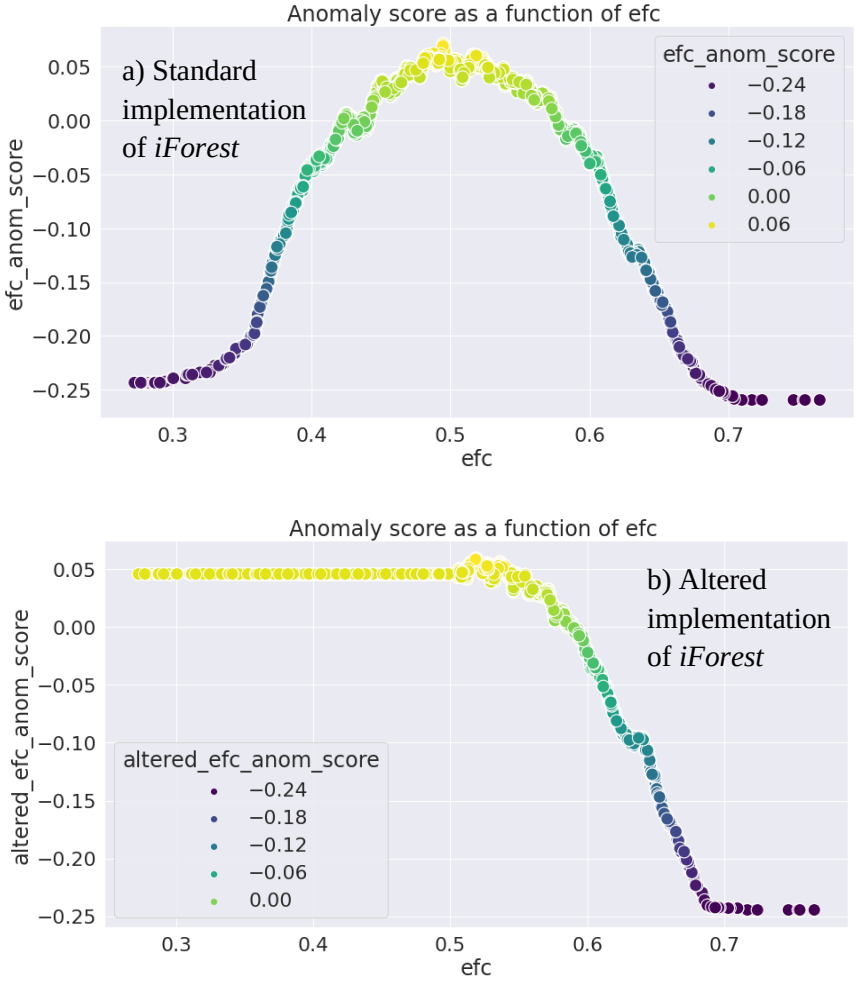


Figure 64. Anomaly score from standard (a) and altered (b) implementation of iForest as a function of "efc" (entropy-focus-criterion) attribute. The darker the point, the higher the anomaly score. (Note: The anomaly score this specific attribute contributes with, not the average anomaly score).

4.2.3 Isolation Forest Results

The quantitative results of the anomaly detection on the different datasets are presented in Table 12 below. The size of the two predicted classes for the different datasets and two explored models are presented. As expected, the altered iForest has a slightly lower number of anomaly predictions. In the BOLD and T1 datasets provided from KI, no datapoint is classified as an outlier. This meets with the expectations, as the datasets have been used in a previous study. In the following paragraphs, the results will be further analyzed and explained by plotting the anomaly score as a function of a few selected features. If nothing else is stated, the method used to extract the anomaly score is altered iForest.

Table 12. Label count in iForest results

Dataset	Standard iForest		Altered iForest	
	Anomaly	**Normal**	**Anomaly**	**Normal**
Subset of MRIQC Database snapshot of BOLD containing meta features	549	11907	443	12013
BOLD data from KI	0	44	0	44
Subset of MRIQC Database snapshot of T1 from 3 Tesla machines (containing meta features)	671	20901	304	21268
T1 data from KI	0	44	0	44

All KI data is classified as normal, and thus have anomaly score greater than 0.However, if we look at anomaly score within the normal class, we can see that deviant datapoints actually get a lower anomaly score, which is what we would expect. See Figure 65, in which the anomaly score is plotted as a function of the feature *aor* (AFNIs outlier ratio). Low values are better. The blue point at the bottom left of the plot gets a lower score than data points with *aor* values close to 0. This pattern is seen in the anomaly score of all BOLD data as well, see Figure 66. The relationship between anomaly score and three other BOLD features, *dvars_std* (low values are good), *fd*_mean (low values are good), and *snr* (high values are good) in all BOLD data is shown in Figure 67, 68, and 69. Similar plots but for the T1 features *cjv* (coefficient for joint vector*)* and *inu_med* (intensity non-uniformity median) are seen in Figure 70 and 71. For *cjv*, higher values are related to the presence of heavy head motion and large INU artifacts, and lower values are thus better. For *inu_med*, values closer to 1.0 are better. The plots show promising results, as seemingly deviant points are classified as anomalous. Figure 65 also shows how the output from iForest can still be used to flag which datapoints at KI have a higher chance of being of worse quality for the dataset.

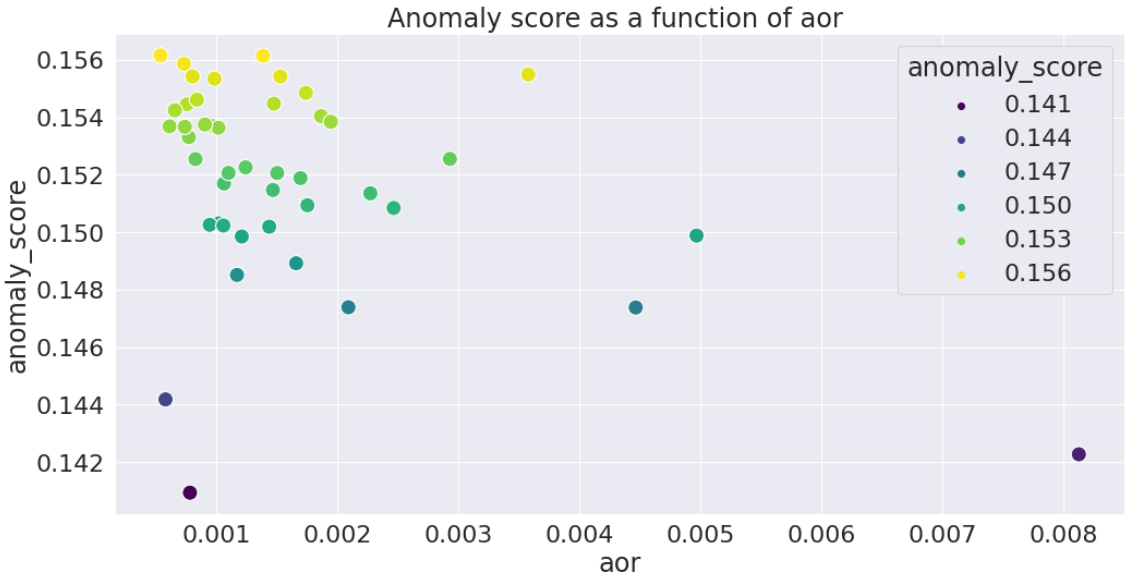


Figure 65. Anomaly score from iForest on KI BOLD data. Datapoints with high AFNIs outlier ratio (aor) values get a lower anomaly score than datapoints with "aor" close to 0. These points can be flagged to researchers at KI to pay additional attention to the quality of the data. The darker the point, the lower the lower the anomaly score.

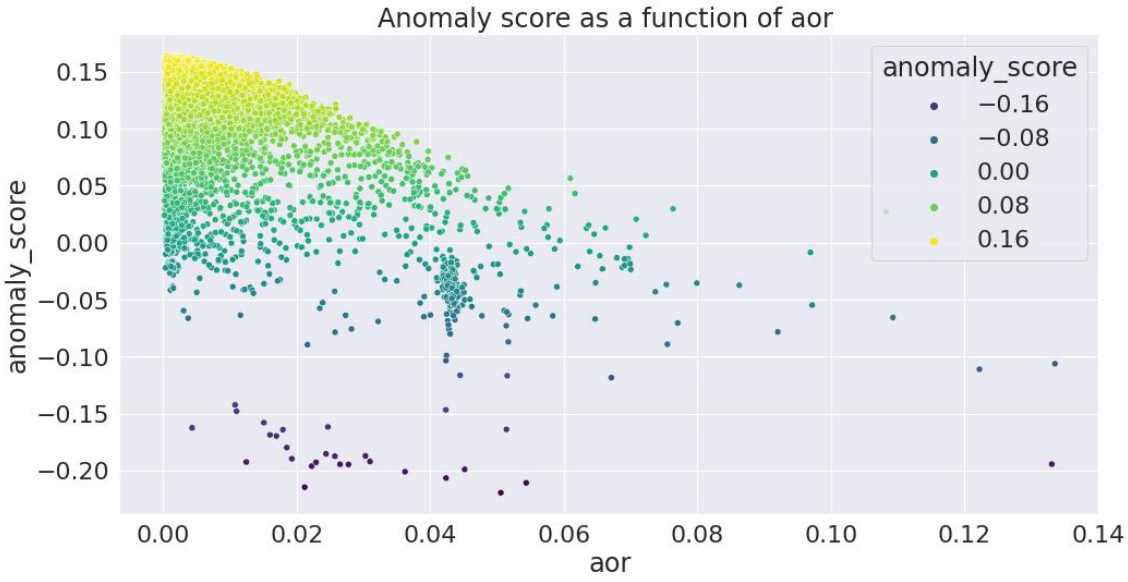


Figure 66. Anomaly scores of all BOLD data. Datapoints with high AFNIs outlier ratio (aor) tend to get a low anomaly score. The darker the point, the lower the anomaly score.

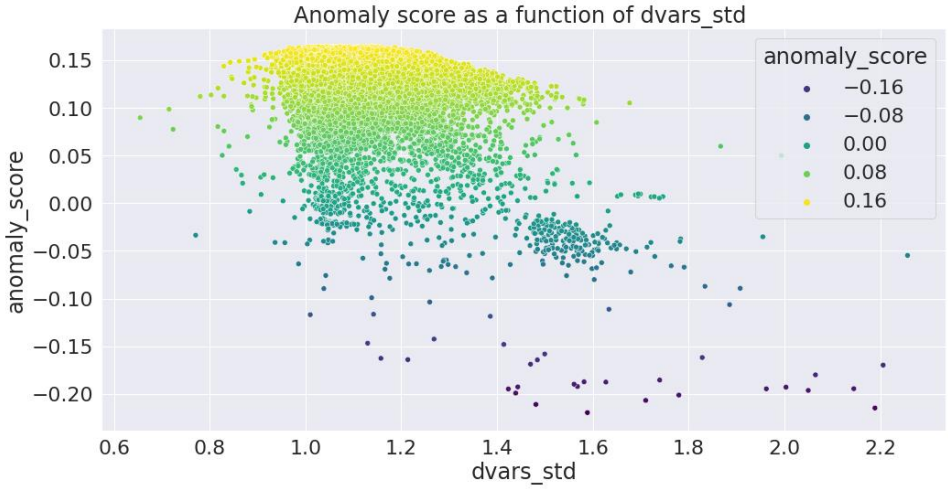


Figure 67. Anomaly scores as a function of "dvars_std" of all BOLD data.

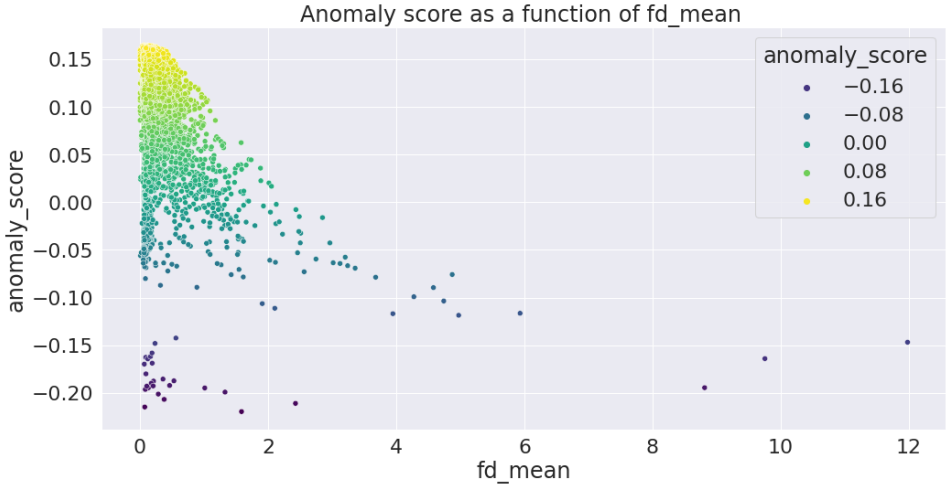


Figure 68. Anomaly scores as a function of "fd_mean" of all BOLD data.

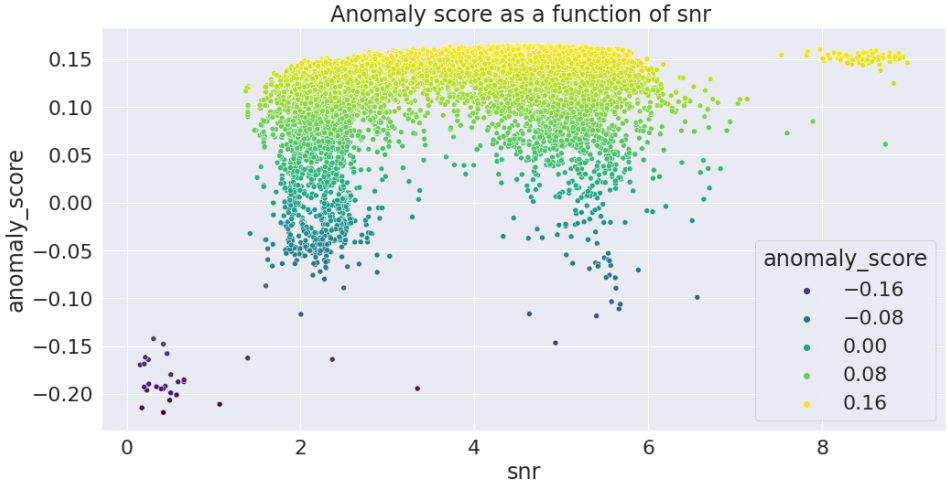


Figure 69. Anomaly scores as a function of "snr" of all BOLD data.

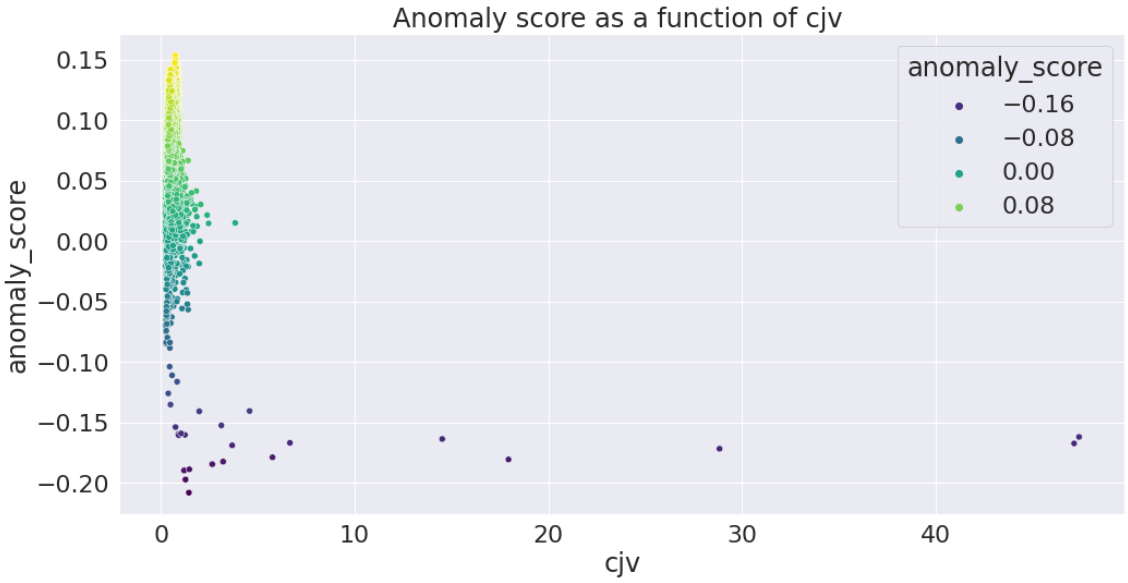


Figure 70. Anomaly scores as a function of "cjv" of all T1 data. The darker the point, the lower the lower the anomaly score. Seemingly deviant points with high "cjv" values are classified as anomalous

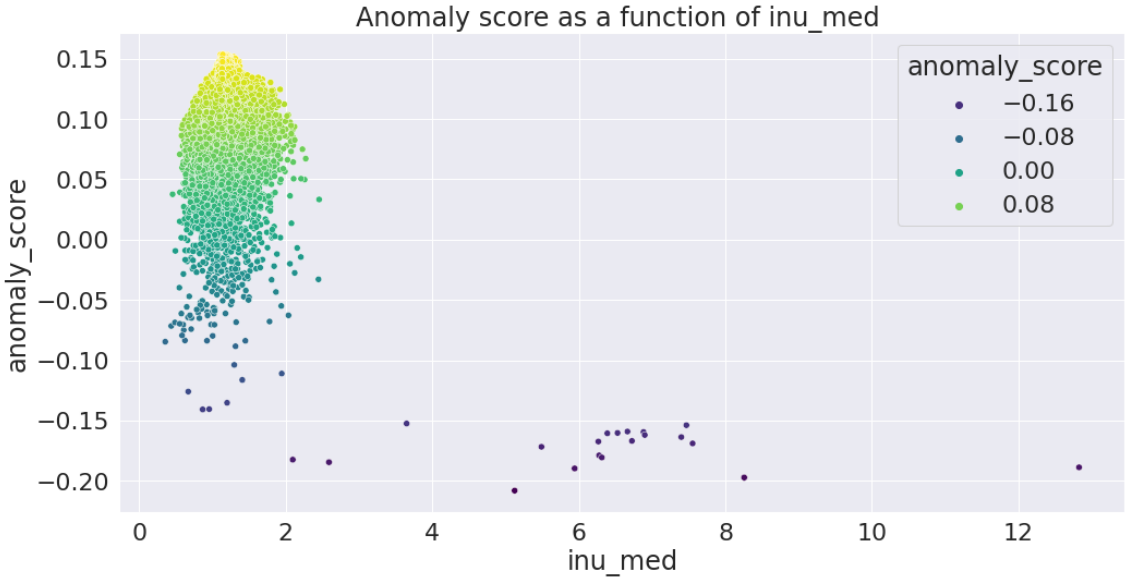


Figure 71. Anomaly scores as a function of "inu_med" of all T1 data. Values around 1 are better. The darker the point, the lower the lower the anomaly score. Seemingly deviant points with high "inu_med" values are classified as anomalous

4.2.4 Relation Between Anomaly Scores and Meta Features

As the IQMs are highly impacted by scanning site and machine model, it is of interest to analyze the relation between these meta attributes and the anomaly score obtained from isolation forest. As previously described in Section 4.2.1 and seen in Figure 53 and Figure 54, a few machines and institution accounts for a large fraction of the data, and there is a possibility data from scanning sites and machines with less support in the database systematically get a higher anomaly score. T-SNE plots of T1 data is seen in Figure 72-75. The same plot is in the different figures colored by anomaly score, label, institution, and machine model. The yellower the dots in plot 72, the more "normal" is the point. The greener, the more anomalous. There undoubtedly seems to be a correlation between normality and support of that meta feature in the dataset. See e.g. how the connected big yellow cluster in the middle of Figure 72, correspond to a widely used machine and highly contributing institution in Figure 74 and 75. Nevertheless, in Figure 73 it can be seen that the points labeled as anomalies are found in the outskirts of the clusters, which is what one intuitively would expect.

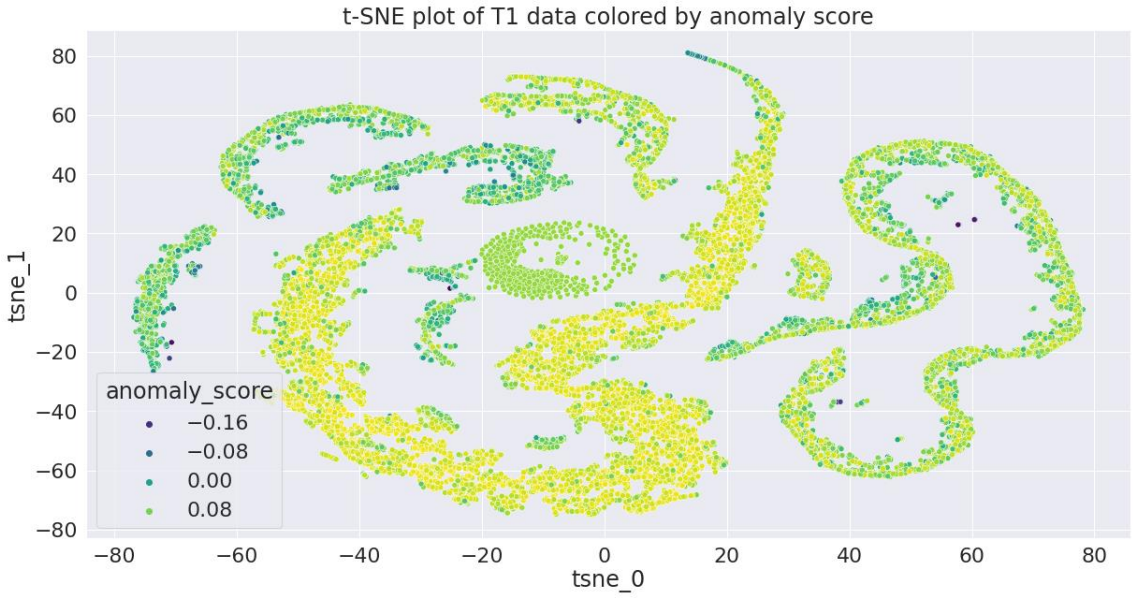


Figure 72. t-SNE plot of T1 data (only from 3T MRI scanners) colored by anomaly score. The higher the score, the darker the point.

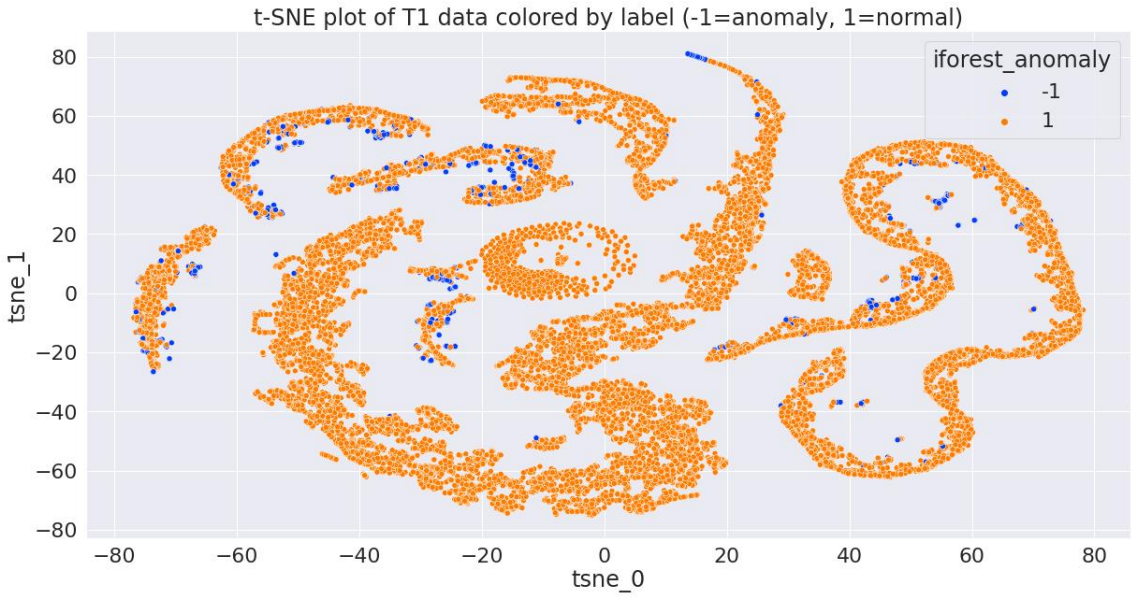


Figure 73. t-SNE plot of T1 data (only from 3T MRI scanners) colored by anomaly label. Normal points are orange, blue points are classified as anomalies

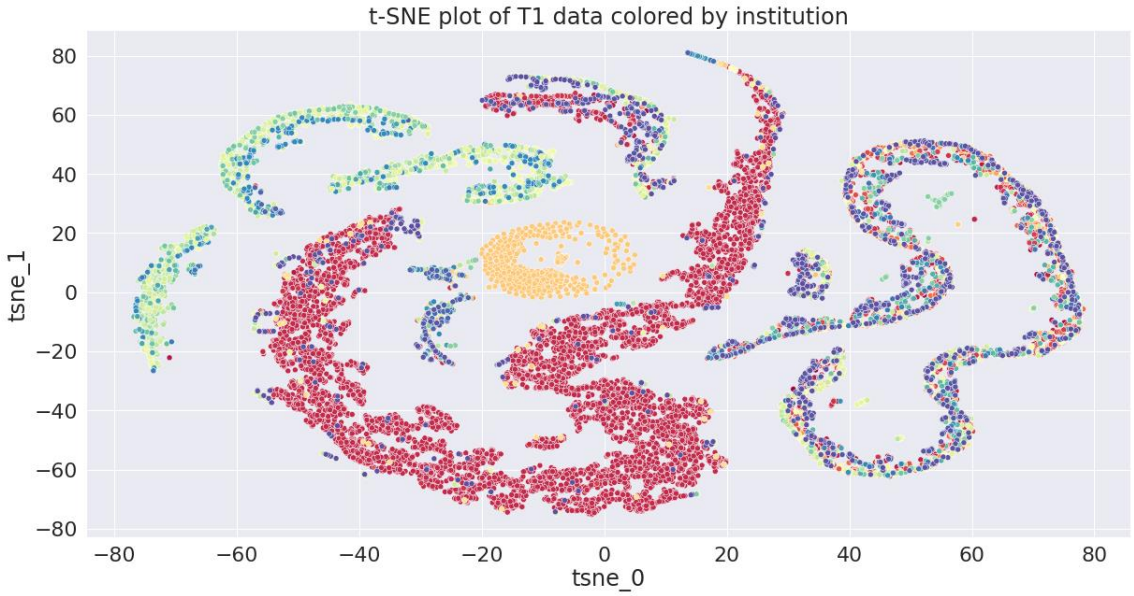


Figure 74. t-SNE plot of T1 data (only from 3T MRI scanners) colored by institution.

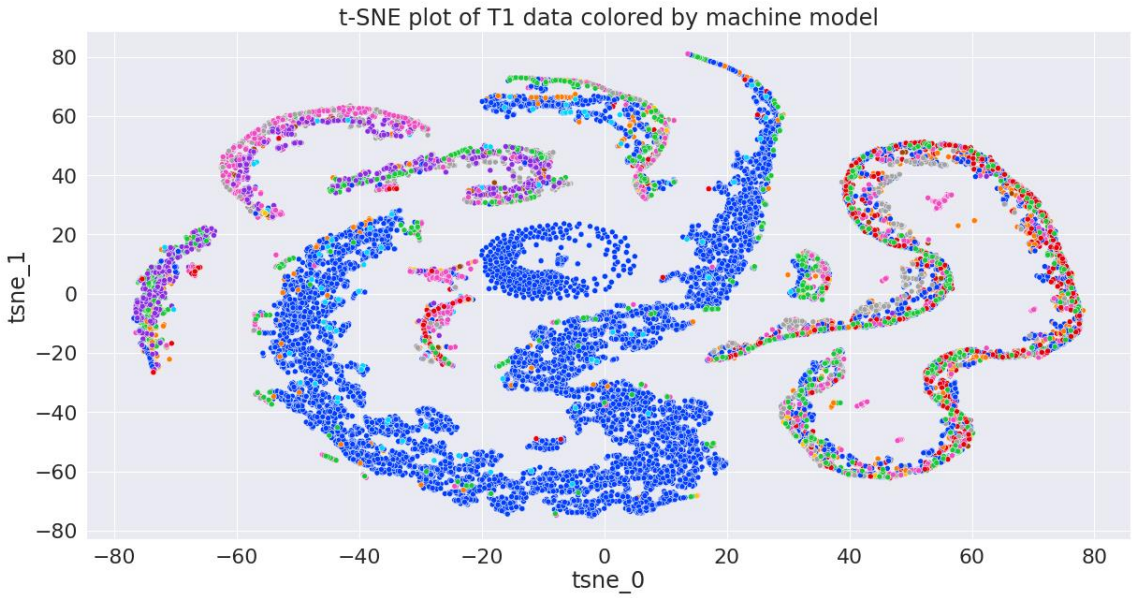


Figure 75. t-SNE plot of T1 data (only from 3T MRI scanners) colored by machine model

5. Discussion

The results indicate that automation of questionnaire transcription is possible, and that anomaly detection of image quality metrics show promising results in identifying deviant data points. It is however clear that full automation is not appropriate, as the risks and costs of faulty predictions outweigh the benefits of such a level of automation. Full automation in systems that do not show sufficient reliability is according to Parasuraman, et al. (2000) only motivated in highly time-critical situations. The context this automated system function is supposed to work in is however not time critical, and full automation is thus not motivated.

In comparison to manual recording of questionnaire results, the *VASReader* system have the potential to significantly reduce the time spent on data transcription. This is especially important in long-term and large-scale projects such as PrePain and will in the long run enable researchers to put more time on their subjects of interest.

A fully automated system that is not totally reliable may cause mistrust among its users, especially if it is black-boxed. We propose that *VASReader* is used as a tool to generate questionnaire result data, which the researchers through visual reports easily can inspect and accept or modify before data is saved for further analysis.

Parasuraman et al. (2000) propose that automation can be applied to four classes of system functions: information acquisition, information analysis, decision selection and action implementation. The anomaly scores and classifications obtained from iForest is a way of automating information analysis. It increases the users' situation awareness and aids decision selection and action implementation (i.e. should the image be kept for further analysis or not?). The anomaly score is an indication of the normality of MRI data from a given scanning session, and low scores imply that the researchers at KI should pay closer attention to the quality of those particular images. Instead of attempting to get an overview of the more than 50 metrics returned from MRIQC, the anomaly score is a concentrated measure of how the quality metrics of a given MRI scan relate to the quality metrics of other MRI scans.

The results of the project's two tasks will be discussed further separately in the following two sections.

5.1 VASReader

VASReader measures VAS marks, recognizes answers to binary questions, and classifies handwritten digits. Although all subtasks are important, the main contribution of this project is the ability to detect marks on VASs. Paper-based VAS questionnaires are extensively used in research. As there, to our knowledge, currently does not exist any research about automatic decoding of VASs, or any tool that offers it, the potential use of *VASReader* extends beyond the PrePain project. *VASReader* provides a systematic and accurate measurement of VASs and offers questionnaire recordings free from inter- and

intra-rater differences. The development of *VASReader* has also resulted in knowledge about machine-reading-enabling document designs. This knowledge is valuable for future design of questionnaires that are supposed to be read using computer vision. It can be concluded that:

- Solid lines are preferred over dashed lines.
- Categorical answers are preferably marked in answer boxes or bubbles, not by encircling an answer alternative.
- If digits are to be recognized in boxes, they need to be written clearly and completely within the bounding box.

If these guidelines are followed, *VASReader* is adaptable to new questionnaire designs after some parameter adjustments.

Future work includes further improvements of each module of *VASReader* and testing the system's performance on larger quantities of test data. The MRI number recognition is one of the modules that needs further work. The performance of different CNN architectures and preprocessing techniques can be evaluated. However, as part of the problem was not in the CNNs predictive ability but that the digits were not contained within the input field, alternative ways of identifying the questionnaires should also be considered. A possible alternative to explore is e.g. attaching a QR code to each questionnaire instead of writing handwritten characters. There already exists reliable techniques for reading QR codes.

VASReader is not yet implemented in the PrePain pipeline, and for it to run in production, it needs to be integrated with the existing IT architecture at the Pain Neuroimaging Lab at KI. Scheduling using *Cron*, containerization with *Singularity* and development of a user interface for handling *VASReader* reports are planned for full pipeline integration. The functionality of *VASReader* has however been presented to the research group, who find the results promising and have expressed a wish to continue improving the tool.

If the methods behind *VASReader* are developed further to be more general and flexible to a greater variance of questionnaire designs, the system's usefulness outside the scope of the PrePain project is enhanced. If more test data is acquired, machine learning approaches are suggested to be explored.

5.2 MRIQC Anomaly detection

The exploration and visualizations of isolation forest anomaly scores indicate that the model is successful in identifying deviant observations. IForest output can thus act as an indicator of predicted quality for future work and be used in the automation process.

The altered implementation of iForest, in which attribute values better than average do not contribute to decrease the average anomaly score has the advantage that scores are

more correlated with, and indicative of, quality. In the standard implementation it is possible that, for two given data points with identical scores, one retrieves it because its IQMs are better than average while the other retrieves it because its IQMs are worse than average. This reduces the situation awareness of the human user, as it may be difficult to deduce the cause of the score. In this way the altered version is more easily interpreted for the researcher who is responsible for quality controlling the MRI data.

There are however shortcomings of the altered version of isolation forest. If a data point has IQMs that are so superior to the rest of the data that it is classified as an outlier by the standard implementation, it could spring from e.g. measurement errors, and for that reason be of interest to take a closer look at. These kinds of deviant data points will not be identified by the altered iForest. In practice, it can thus be beneficial to take into account the scores from both versions of iForest.

One important remark is that what this iForest model identifies are anomalies in image quality metrics from MRI data. It is not applied directly to the image data, and the relationship between anomaly scores in IQMs and actual image quality has not been investigated in this thesis project. If substandard image quality is not reflected in the IQMs, subpar images will not be detected using the implemented anomaly detection model. Vice versa, if deviant IQMs do not reflect bad image quality, anomaly scores are misleading. There is reason to further investigate the relation between IQMs and actual image quality, as previous research has given reason to question IQMs ability to reflect actual image quality. As a reminder, it was stated in section 2.3.2 that anomaly detection with iForest is a viable method for the detection of anomalous instances if i) substandard images are assumed to be few and different from most of the data, and ii) this is reflected in the IQMs. For this reason, iForest anomaly scores should function as an indication of the image quality and is suggested to be utilized as way flagging images that need to be payed closer attention during visual inspection. It is thus an automated form of information analysis, helping the researcher in decision making.

As seen in the t-SNE plots in section 4.2.4, data from machines and institutions with high support in the data set are in general scored as more normal. The fact that there are clear systematic differences in IQMs between sites makes data from less frequently occurring or unseen sites and machines more probable to be susceptible to isolation. This does not necessarily mean the actual quality is worse, as the variability in IQMs may reflect the MRI machine or institution more than the actual quality.

IQMs are different types of, from imaging data, extracted quality measures. The IQMs are extracted as an efficient way to summarize information about quality in image data. Presumably, much information in the complex 3D and 4D images is lost when they are boiled down to a set of metrics, and it is likely that better quality prediction can be achieved by machine learning models that take images as input, e.g. CNNs. These are specialized in handling the spatial topology in images. Implementation of CNNs though impose completely different requirements on time and memory compared to training a machine learning model on IQMs.

Future work includes comparing iForest with other machine learning algorithms and training the models larger quantities of MRIQC data from the MRIQC WebAPI. It is also of interest to further explore the relation between IQMs, anomaly scores and meta features other than the ones selected in this project. If more ratings are uploaded to the MRIQC database, supervised learning approaches are recommended to be explored.

6. Conclusions

The overarching aim of this project has been to investigate whether time-consuming and error-prone manual tasks within cognitive neuroscience research can be automated. Specifically, two subtasks within the broader scope of automatizing research data pipelines have been addressed. The first task has been to investigate the possibility of automatic transcription of questionnaire data. The second task has been to implement and evaluate the performance of an anomaly detection method trained on MRIQC data.

The results show that it is possible to reliably decode questionnaires containing visual analog scales. With inspiration from previous work within image document processing, OMR and ICR, a computer vision system, called *VASReader*, has been developed. *VASReader* is built specifically for VAS questionnaires and is, to our knowledge, the first system to address the specific case of decoding VASs.

It can also be concluded that the unsupervised machine learning algorithm *Isolation Forest* shows promising results in classifying MRIQC data as anomalous or normal. The retrieved anomaly scores reflect the underlying distribution of the image quality metric (IQM) features, and visualizations of results show that apparent deviant datapoints are correctly classified as anomalies. Further research is however needed to determine the relation between anomalous IQMs and actual image quality. Due to strong site-effects in the IQMs, the model is also prone to classify data from less frequently seen sites or unseen sites as more anomalous than data from MRI machines and institutions with high support in the MRIQC database.

To conclude, it is evident that manual tasks within cognitive science research can be automated, and that there is much to gain from automation. However, the project illustrates the importance of selecting an appropriate type and level of automation, and full automation is not the suggested option for either of the above described tasks.

References

Afifi, M. and Hussain, K. (2017) 'The Achievement of Higher Flexibility in Multiple Choice-based Tests Using Image Classification Techniques', *International Journal on Document Analysis and Recognition (IJDAR)*. doi: 10.1007/s10032-019-00322-3.

Ahlawat, S. *et al.* (2020) 'Improved Handwritten Digit Recognition Using Convolutional Neural Networks (CNN)', *Sensors*, 20(3344), p. 3344. doi: 10.3390/s20123344.

Bainbridge, L. (1983) 'Ironies of automation', *Automatica*, 19(6), pp. 775–779. doi: 10.1016/0005-1098(83)90046-8.

Barnett, V. and Lewis, T. (1994) *Outliers in statistical data*. 3. ed. Chichester: Wiley (Wiley series in probability and mathematical statistics).

Barney Smith, E., Nagy, G. and Lopresti, D. (2009) 'Mark Detection from Scanned Ballots.', in. *Proceedings of SPIE - The International Society for Optical Engineering*, pp. 1–10. doi: 10.1117/12.806468.

Boudraa, O., Hidouci, W. K. and Michelucci, D. (2020) 'Using skeleton and Hough transform variant to correct skew in historical documents', *Mathematics and Computers in Simulation*, 167, pp. 389–403. doi: 10.1016/j.matcom.2019.05.009.

Butterfield, A. B., Ngondi, G. E. N. E. and Kerr, A. K. (2016) 'OCR', in Butterfield, A., Ngondi, G. E., and Kerr, A. (eds) *A Dictionary of Computer Science*. Oxford University Press. Available at: http://www.oxfordreference.com/view/10.1093/acref/9780199688975.001.0001/acref-9780199688975-e-3589 (Accessed: 28 October 2020).

Chai, D. (2016) 'Automated marking of printed multiple choice answer sheets', *2016 IEEE International Conference on Teaching, Assessment, and Learning for Engineering (TALE)*. doi: 10.1109/TALE.2016.7851785.

Chouvatut, V. and Prathan, S. (2014) 'The flexible and adaptive X-mark detection for the simple answer sheets', in *2014 International Computer Science and Engineering Conference (ICSEC)*. *2014 International Computer Science and Engineering Conference (ICSEC)*, pp. 433–439. doi: 10.1109/ICSEC.2014.6978236.

Delgado, D. A. *et al.* (2018) 'Validation of Digital Visual Analog Scale Pain Scoring With a Traditional Paper-based Visual Analog Scale in Adults', *Journal of the American Academy of Orthopaedic Surgeons. Global Research & Reviews*, 2(3). doi: 10.5435/JAAOSGlobal-D-17-00088.

Deng, L. (2012) 'The MNIST Database of Handwritten Digit Images for Machine Learning Research [Best of the Web]', *Signal Processing Magazine, IEEE*, 29, pp. 141–142. doi: 10.1109/MSP.2012.2211477.

Dietrich, O. *et al.* (2007) 'Measurement of signal-to-noise ratios in MR images: Influence of multichannel coils, parallel imaging, and reconstruction filters', *Journal of Magnetic Resonance Imaging*, 26(2), pp. 375–385. doi: https://doi.org/10.1002/jmri.20969.

Douglas, D. and Peucker, T. (1973) 'ALGORITHMS FOR THE REDUCTION OF THE NUMBER OF POINTS REQUIRED TO REPRESENT A DIGITIZED LINE OR ITS CARICATURE'. doi: 10.3138/FM57-6770-U75U-7727.

Elias, E. M. de, Tasinaffo, P. M. and Hirata, R. (2019) 'Alignment, Scale and Skew Correction for Optical Mark Recognition Documents Based', in *2019 XV Workshop de Visão Computacional (WVC)*. *2019 XV Workshop de Visão Computacional (WVC)*, pp. 26–31. doi: 10.1109/WVC.2019.8876933.

Erasmus, L. J. *et al.* (2004) 'A short overview of MRI artefacts', *South African Journal of Radiology*, 8(2), p. 13. doi: 10.4102/sajr.v8i2.127.

Esteban, O. *et al.* (2017) 'MRIQC: Advancing the automatic prediction of image quality in MRI from unseen sites.', *PLoS ONE*, 12(9), p. e0184661. doi: 10.1371/journal.pone.0184661.

Esteban, O. *et al.* (2018) *Crowdsourced MRI quality metrics and expert quality annotations for training of humans and machines*. doi: 10.1101/420984.

Esteban, O. *et al.* (2019) 'Crowdsourced MRI quality metrics and expert quality annotations for training of humans and machines', *Scientific Data*, 6(1), p. 30. doi: 10.1038/s41597-019-0035-4.

Esteban, O., Poldrack, R. A. and Gorgolewski, K. J. (2018) 'Improving Out-of-Sample Prediction of Quality of MRIQC', in Stoyanov, D. et al. (eds) *Intravascular Imaging and Computer Assisted Stenting and Large-Scale Annotation of Biomedical Data and Expert Label Synthesis*. Cham: Springer International Publishing (Lecture Notes in Computer Science), pp. 190–199. doi: 10.1007/978-3-030-01364-6_21.

Fisteus, J. A., Pardo, A. and García, N. F. (2013) 'Grading Multiple Choice Exams with Low-Cost and Portable Computer-Vision Techniques', *Journal of Science Education and Technology*, 22(4), pp. 560–571. doi: 10.1007/s10956-012-9414-8.

Funke, F. and Reips, U.-D. (2006) 'Visual analogue scales in online surveys: non-linear data categorization by transformation with reduced extremes', *8th International Conference GOR06, March*, pp. 21–22.

Hahn, E. C. and Hansman, R. (2013) 'An Experimental Study of the Effects of Automation on Pilot Situational Awareness in the Datalink ATC Environment', in. doi: 10.4271/922022.

Haskins, B. (2015) 'Contrasting classifiers for software-based OMR responses', in, pp. 233–238. doi: 10.1109/RoboMech.2015.7359528.

Hawkins, D. (1980) *Identification of Outliers*. Springer Netherlands (Monographs on Statistics and Applied Probability). doi: 10.1007/978-94-015-3994-4.

Hirata, D. and Takahashi, N. (2020) *Ensemble learning in CNN augmented with fully connected subnetworks*.

Klimek, L. *et al.* (2017) 'Visual analogue scales (VAS): Measuring instruments for the documentation of symptoms and therapy monitoring in cases of allergic rhinitis in everyday health care', *Allergo Journal International*, 26(1), pp. 16–24. doi: 10.1007/s40629-016-0006-7.

Krüger, G. and Glover, G. H. (2001) 'Physiological noise in oxygenation-sensitive magnetic resonance imaging', *Magnetic Resonance in Medicine*, 46(4), pp. 631–637. doi: 10.1002/mrm.1240.

Lecun, Y. *et al.* (1998) 'Gradient-based learning applied to document recognition', *Proceedings of the IEEE*, 86(11), pp. 2278–2324. doi: 10.1109/5.726791.

Lee, J. D. and See, K. A. (2004) 'Trust in Automation: Designing for Appropriate Reliance', *Human Factors*, 46(1), pp. 50–80. doi: 10.1518/hfes.46.1.50_30392.

de Lima, O. *et al.* (2016) 'Signature line detection in scanned documents', in, pp. 3254–3258. doi: 10.1109/ICIP.2016.7532961.

Liu, F. T., Ting, K. M. and Zhou, Z. (2008) 'Isolation Forest', in *2008 Eighth IEEE International Conference on Data Mining*. *2008 Eighth IEEE International Conference on Data Mining*, pp. 413–422. doi: 10.1109/ICDM.2008.17.

Liu, F. T., Ting, K. and Zhou, Z.-H. (2012) 'Isolation-Based Anomaly Detection', *ACM Transactions on Knowledge Discovery From Data - TKDD*, 6, pp. 1–39. doi: 10.1145/2133360.2133363.

Liu, W., Wei, J. and Meng, Q. (2020) 'Comparisions on KNN, SVM, BP and the CNN for Handwritten Digit Recognition', in *2020 IEEE International Conference on Advances in Electrical Engineering and Computer Applications(AEECA)*. *2020 IEEE International Conference on Advances in Electrical Engineering and Computer Applications(AEECA)*, pp. 587–590. doi: 10.1109/AEECA49918.2020.9213482.

Loke, S. C., Kasmiran, K. A. and Haron, S. A. (2018) 'A new method of mark detection for software-based optical mark recognition', *PLOS ONE*, 13(11), p. e0206420. doi: 10.1371/journal.pone.0206420.

Marsh-Richard, D. M. *et al.* (2009) 'Adaptive Visual Analog Scales (AVAS): A Modifiable Software Program for the Creation, Administration, and Scoring of Visual Analog Scales', *Behavior research methods*, 41(1), pp. 99–106. doi: 10.3758/BRM.41.1.99.

Matas, J., Galambos, C. and Kittler, J. (2000) 'Robust Detection of Lines Using the Progressive Probabilistic Hough Transform', *Computer Vision and Image Understanding*, 78, pp. 119–137. doi: 10.1006/cviu.1999.0831.

MNIST handwritten digit database, Yann LeCun, Corinna Cortes and Chris Burges (no date). Available at: http://yann.lecun.com/exdb/mnist/ (Accessed: 21 October 2020).

Mordvintsev, A. and Abid, K. (2014) 'Opencv-python tutorials documentation', *Obtenido de https://media. readthedocs. org/pdf/opencv-python-tutroals/latest/opencv-python-tutroals. pdf*.

Mortamet, B. *et al.* (2009) 'Automatic quality assessment in structural brain magnetic resonance imaging', *Magnetic Resonance in Medicine*, 62(2), pp. 365–372. doi: https://doi.org/10.1002/mrm.21992.

'MRIQC WebAPI - Database snapshot' (2019). figshare. doi: 10.6084/m9.figshare.7097879.v4.

Mukhopadhyay, P. and Chaudhuri, B. B. (2015) 'A survey of Hough Transform', *Pattern Recognition*, 48(3), pp. 993–1010. doi: 10.1016/j.patcog.2014.08.027.

Nulty, D. D. (2008) 'The adequacy of response rates to online and paper surveys: what can be done?', *Assessment & Evaluation in Higher Education*. doi: 10.1080/02602930701293231.

OpenCV (2020a) *Camera Calibration and 3D Reconstruction*. Available at: https://docs.opencv.org/master/d9/d0c/group__calib3d.html#ga4abc2ece9fab9398f2e560d53c8c9780 (Accessed: 4 November 2020).

OpenCV (2020b) *Eroding and Dilating*. Available at: https://docs.opencv.org/3.4/db/df6/tutorial_erosion_dilatation.html (Accessed: 24 November 2020).

OpenCV (2020c) *Feature Detection*. Available at: https://docs.opencv.org/4.5.0/dd/d1a/group__imgproc__feature.html#ga861818 0a5948286384e3b7ca02f6feeb (Accessed: 20 October 2020).

OpenCV (2020d) *Feature Matching + Homography to find Objects*. Available at: https://docs.opencv.org/master/d1/de0/tutorial_py_feature_homography.html (Accessed: 19 November 2020).

OpenCV (2020e) *Geometric Image Transformations*. Available at: https://docs.opencv.org/master/da/d54/group__imgproc__transform.html#gaf73673a7e8e18ec6963e3774e6a94b87 (Accessed: 20 November 2020).

OpenCV (2020f) *Hough Line Transform*. Available at: https://docs.opencv.org/4.5.0/d9/db0/tutorial_hough_lines.html (Accessed: 20 October 2020).

OpenCV: Basic Thresholding Operations (no date). Available at: https://docs.opencv.org/3.4/db/d8e/tutorial_threshold.html (Accessed: 24 November 2020).

OpenCV: Image Thresholding (no date). Available at: https://docs.opencv.org/master/d7/d4d/tutorial_py_thresholding.html (Accessed: 24 November 2020).

OpenCV: Structural Analysis and Shape Descriptors (2018). Available at: https://docs.opencv.org/4.0.0/d3/dc0/group__imgproc__shape.html#gadf1ad6a0b82947fa1fe3c3d497f260e0 (Accessed: 4 November 2020).

Parasuraman, R., Sheridan, T. B. and Wickens, C. D. (2000) 'A model for types and levels of human interaction with automation', *IEEE transactions on systems, man, and cybernetics. Part A, Systems and humans: a publication of the IEEE Systems, Man, and Cybernetics Society*, 30(3), pp. 286–297. doi: 10.1109/3468.844354.

PCP Quality Assessment Protocol (no date). Available at: http://preprocessed-connectomes-project.org/quality-assessment-protocol/ (Accessed: 1 December 2020).

Power, J. D. *et al.* (2012) 'Spurious but systematic correlations in functional connectivity MRI networks arise from subject motion', *Neuroimage*, 59(3), pp. 2142–2154. doi: 10.1016/j.neuroimage.2011.10.018.

Power, J. D., Schlaggar, B. L. and Petersen, S. E. (2015) 'Recent progress and outstanding issues in motion correction in resting state fMRI', *NeuroImage*, 0, pp. 536–551. doi: 10.1016/j.neuroimage.2014.10.044.

Ptucha, R. *et al.* (2019) 'Intelligent character recognition using fully convolutional neural networks', *Pattern Recognition*, 88, pp. 604–613. doi: 10.1016/j.patcog.2018.12.017.

Reips, U.-D. and Funke, F. (2008) 'Interval-level measurement with visual analogue scales in Internet-based research: VAS Generator', *Behavior Research Methods*, 40(3), pp. 699–704. doi: 10.3758/BRM.40.3.699.

Rublee, E. *et al.* (2011) 'ORB: An efficient alternative to SIFT or SURF', in *2011 International Conference on Computer Vision*. *2011 International Conference on Computer Vision*, pp. 2564–2571. doi: 10.1109/ICCV.2011.6126544.

Running mriqc — mriqc 0.1 documentation (no date). Available at: https://mriqc.readthedocs.io/en/latest/running.html#command-line-interface (Accessed: 9 November 2020).

S, R., Atal, K. and Arora, A. (2013) 'Cost Effective Optical Mark Reader', *International Journal of Computer Science and Artificial Intelligence*, 3, pp. 44–49. doi: 10.5963/IJCSAI0302002.

Sattayakawee, N. (2013) 'Test Scoring for Non-Optical Grid Answer Sheet Based on Projection Profile Method', *International Journal of Information and Education Technology*, pp. 273–277. doi: 10.7763/IJIET.2013.V3.279.

Shah, L. *et al.* (2010) 'Functional Magnetic Resonance Imaging', *Seminars in roentgenology*, 45, pp. 147–56. doi: 10.1053/j.ro.2009.09.005.

Sheridan, T., Verplank, W. and Brooks, T. (1978) 'Human and Computer Control of Undersea Teleoperators'.

Suzuki, S. and Abe, K. (1985) 'Topological structural analysis of digitized binary images by border following', *Comput. Vis. Graph. Image Process.* doi: 10.1016/0734-189X(85)90016-7.

Tukey, J. W. (1977) *Exploratory data analysis*. Reading, Mass: Addison-Wesley (Book, Whole).

Van Dijk, K. R. A., Sabuncu, M. R. and Buckner, R. L. (2012) 'The Influence of Head Motion on Intrinsic Functional Connectivity MRI', *NeuroImage*, 59(1), pp. 431–438. doi: 10.1016/j.neuroimage.2011.07.044.

Yan Ping Zhou and Chew Lim Tan (2000) 'Hough technique for bar charts detection and recognition in document images', in *Proceedings 2000 International Conference on Image Processing (Cat. No.00CH37101)*. *Proceedings 2000 International Conference on Image Processing (Cat. No.00CH37101)*, pp. 605–608 vol.2. doi: 10.1109/ICIP.2000.899506.

Yefeng, Z., Huiping, L. and Doermann, D. (2005) 'A parallel-line detection algorithm based on HMM decoding', *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(5), pp. 777–792. doi: 10.1109/TPAMI.2005.89.

Appendix A

A 1. First version of the PrePain Questionnaire. Page 1.

Smärta just nu

- Har du smärta någonstans I kroppen just nu? JA NEJ
  Om JA, var? Vänligen sätt ett tydligt kryss och välj intensitet:

	Ingen smärta	**Värta tänkbara smärta**
Huvudet	[----------]	
Käken	[----------]	
Nacken	[----------]	
Ryggen	[----------]	
Axlarna	[----------]	
Halsen	[----------]	
Magen	[----------]	
Armar	[----------]	
Ben	[----------]	
Höft	[----------]	
Fötter	[----------]	

Långvarig smärta

Långvarig smärta är den typ av smärta som inte går att förklara med tillfällig huvudvärk, träningsvärk, mensvärk eller liknande. Långvarig smärta varar i mer än tre månader i sträck.

- Lider du just nu av långvarig smärta? JA NEJ
- Har du tidigare haft perioder med mycket smärta i kroppen? JA NEJ
- Har du någon i din nära familj som lider av långvarig smärta? JA NEJ
- Är du orolig för att drabbas av långvarig smärta? JA NEJ

Hur brukar det vara

- Hur känslig upplever du att du är för smärta jämfört med andra, 0-100 mm:
  Sätt ett tydligt kryss på den streckade linjen.

[----------]

Väldigt smärtkänslig	Normal	Väldigt smärttålig

Vänligen vänd sida!

A.2 First version of the PrePain Questionnaire. Page 2.

- I vilken grad undviker du situationer på grund av risk för att få ont (t.ex., tatueringar, fotboll, tandläkarbesök):

  [----------]

  Alltid Ibland Aldrig

- Hur jobbigt tycker du att det är när du har ont?

  [----------]

  Väldigt jobbigt Inte alls jobbigt

- Hur lättad känner du dig när smärtan försvinner?

  [----------]

  Väldigt lättad Inte alls

- I allmänhet hur mycket fokuserar du på din kropp och kroppens reaktioner när du tror att du är sjuk?

  [----------]

  Väldigt mycket Inte alls

- Brukar smärta kunna leda till negativa tankar om att din hälsa håller på att förvärras?

  [----------]

  Alltid Ibland Aldrig

Appendix B

B.1 Second version of the PrePain Questionnaire. Page 1.

MR-nummer (fylls ej i av deltagare)				

Smärta just nu

- Har du smärta någonstans i kroppen just nu? ☐JA ☐NEJ
  Om JA, var? Vänligen sätt ett tydligt kryss och välj intensitet:

	Ingen smärta	**Värsta tänkbara smärta**
Huvudet	[———————————————]	
Käken	[———————————————]	
Nacken	[———————————————]	
Ryggen	[———————————————]	
Axlarna	[———————————————]	
Halsen	[———————————————]	
Magen	[———————————————]	
Armar	[———————————————]	
Ben	[———————————————]	
Höft	[———————————————]	
Fötter	[———————————————]	

Långvarig smärta

Långvarig smärta är den typ av smärta som inte går att förklara med tillfällig huvudvärk, träningsvärk, mensvärk eller liknande. Långvarig smärta varar i mer än tre månader i sträck.

- Lider du just nu av långvarig smärta? ☐JA ☐NEJ
- Har du tidigare haft perioder med mycket smärta i kroppen? ☐JA ☐NEJ
- Har du någon i din nära familj som lider av långvarig smärta? ☐JA ☐NEJ
- Är du orolig för att drabbas av långvarig smärta? ☐JA ☐NEJ

Hur brukar det vara

- Hur känslig upplever du att du är för smärta jämfört med andra, 0-100 mm:
  Sätt ett tydligt kryss på linjen.

[———————————————]

Väldigt smärtkänslig — Normal — Väldigt smärttålig

Vänligen vänd sida!

B.2 Second version of the PrePain Questionnaire. Page 2.

- I vilken grad undviker du situationer på grund av risk för att få ont (t.ex., tatueringar, fotboll, tandläkarbesök):

[———————————————————————————]

Alltid Ibland Aldrig

- Hur jobbigt tycker du att det är när du har ont?

[———————————————————————————]

Väldigt jobbigt Inte alls jobbigt

- Hur lättad känner du dig när smärtan försvinner?

[———————————————————————————]

Väldigt lättad Inte alls

- I allmänhet hur mycket fokuserar du på din kropp och kroppens reaktioner när du tror att du är sjuk?

[———————————————————————————]

Väldigt mycket Inte alls

- Brukar smärta kunna leda till negativa tankar om att din hälsa håller på att förvärras?

[———————————————————————————]

Alltid Ibland Aldrig

Appendix C

Table 13. IQM features for structural (T1w) images

Feature	**Data type**	**Interpretation** (Esteban *et al.*, 2017)
Measures based on noise measurements		
cjv	float64	Coefficient of joint variation. Higher values are related to the presence of heavy head motion and large INU artifacts. Lower values are better.
cnr	float64	Contrast-to-noise ratio. Extension of the SNR calculation to evaluate how separated the tissue distributions of GM and WM are. Higher values indicate better quality.
snr*	float64	Signal-to-noise-ratio. Calculated within the tissue mask
snrd**	float64	Dietrich's SNR (SNRd) as proposed by (Dietrich *et al.*, 2007)
qi_2	float64	Mortamet's quality index 2 (Mortamet *et al.*, 2009)
Measures based on information theory		
efc	float64	Uses the Shannon entropy of voxel intensities as an indication of ghosting and blurring induced by head motion. Lower values are better.
fber	float64	Defined as the mean energy of image values within the head relative to outside the head(*PCP Quality Assessment Protocol*, no date). Higher values are better.
Measures targeting specific artifacts		
inu_***	float64	Summary statistics (max, min and median) of the INU field as extracted by the N4ITK algorithm. Values closer to 1.0 are better.
qi_1	float64	The QI1 is the proportion of voxels with intensity corrupted by artifacts normalized by the number of voxels in the background. Lower values are better.

Cont. on next page

wm2max	float64	The white matter to maximum intensity ratio is the median intensity within the WM mask over the 95% percentile of the full intensity distribution, that captures the existence of long tails due to hyper-intensity of the carotid vessels and fat. Values should be around the interval [0.6, 0.8].

Other measures

fwhm****	float64	The FWHM of the spatial distribution of the image intensity values in units of voxels. Lower values are better.
icvs_*****	float64	The ICV fractions of CSF, GM and WM. They should move within a normative range.
rpve_******	float64	The rPVe of CSF, GM and WM. Lower values are better.
summary_stats*******	float64	Mean, standard deviation, 5% percentile and 95% percentile of the distribution of background, CSF, GM and WM.
tpm_overlap********	float64	The overlap of the TPMs estimated from the image and the corresponding maps from the ICBM nonlinear-asymmetric 2009c template

*	'snr_csf', 'snr_gm', 'snr_total', 'snr_wm',
**	'snrd_csf', 'snrd_gm', 'snrd_total', 'snrd_wm',
***	'inu_med', 'inu_range',
****	'fwhm_avg', 'fwhm_x', 'fwhm_y', 'fwhm_z',
*****	'icvs_csf', 'icvs_gm', 'icvs_wm',
******	'rpve_csf', 'rpve_gm', 'rpve_wm',
*******	'summary_bg_k', 'summary_bg_mad', 'summary_bg_mean', 'summary_bg_median', 'summary_bg_n', 'summary_bg_p05', 'summary_bg_p95', 'summary_bg_stdv', 'summary_csf_k', 'summary_csf_mad', 'summary_csf_mean', 'summary_csf_median', 'summary_csf_n', 'summary_csf_p05', 'summary_csf_p95', 'summary_csf_stdv', 'summary_gm_k', 'summary_gm_mad', 'summary_gm_mean', 'summary_gm_median', 'summary_gm_n', 'summary_gm_p05', 'summary_gm_p95', 'summary_gm_stdv', 'summary_wm_k', 'summary_wm_mad', 'summary_wm_mean', 'summary_wm_median', 'summary_wm_n', 'summary_wm_p05', 'summary_wm_p95', 'summary_wm_stdv',
********	'tpm_overlap_csf', 'tpm_overlap_gm', 'tpm_overlap_wm',

Appendix D

Table 14. IQM features for functional (BOLD) images

Feature	**Data type**	**Interpretation** (Esteban *et al.*, 2017)
Measures for the spatial information		
efc	float64	Entropy-focus criterion
fber	float64	Foreground-Background energy ratio
fwhm_*	float64	Full-width half maximum smoothness
snr	float64	Signal-to-noise-ratio
summary_stats**	float64	Estimates the mean, the standard deviation, the 95% and the 5% percentiles of each tissue distribution.
Measures for the temporal information		
dvars_ ***	float64	D referring to temporal derivative of time courses, VARS referring to RMS variance over voxels
gcor	float64	Global Correlation
tsnr	float64	Temporal SNR, a simplified interpretation of the tSNR definition (Krüger and Glover, 2001)
Measures for artifacts and other		
fd_****	float64	Framewise Displacement. Expresses instantaneous head-motion
gsr_*****	float64	Ghost to Signal Ratio
aor	float64	AFNI's outlier ratio. Mean fraction of outliers per fMRI volume
aqi	float64	AFNI’s quality index

* 'fwhm_x', 'fwhm_y', 'fwhm_z'

** 'summary_bg_k', 'summary_bg_mad', 'summary_bg_mean', 'summary_bg_median', 'summary_bg_p05', 'summary_bg_p95', 'summary_bg_stdv', 'summary_fg_k', 'summary_fg_mad', 'summary_fg_mean', 'summary_fg_median', 'summary_fg_n', 'summary_fg_p05', 'summary_fg_p95', 'summary_fg_stdv'

*** 'dvars_nstd', 'dvars_std', 'dvars_vstd'

**** 'fd_mean', 'fd_num', 'fd_perc'

**** 'gsr_x', 'gsr_y'

Appendix E

Table 15. Full List of Selected Features

Selected T1 IQM Features	Selected BOLD IQM Features
'cjv'	'aqi'
'efc'	'aor'
'fber'	'dvars_nstd'
'fwhm_avg'	'dvars_std'
'icvs_csf'	'dvars_vstd'
'icvs_wm'	'efc'
'inu_med'	'fber'
'inu_range'	'fd_mean'
'qi_1'	'fd_num'
'qi_2'	'fwhm_avg'
'rpve_csf'	'gcor'
'snr_csf'	'gsr_x'
'snr_gm'	'gsr_y'
'summary_bg_k'	'snr'
'summary_bg_n'	'summary_bg_k'
'summary_csf_k'	'summary_bg_k'
'summary_csf_mad'	'summary_bg_mad'
'summary_csf_mean'	'summary_bg_n'
'summary_csf_n'	'summary_fg_k'
'summary_csf_p05'	'tsnr'
'summary_gm_k'	
'summary_gm_mean'	
'summary_gm_n'	
'summary_wm_k'	
'summary_wm_mean'	
'summary_wm_median'	
'tpm_overlap_csf'	
'tpm_overlap_wm'	
'wm2max'	

Appendix F

Figure 76. CNN architecture used for handwritten digit recognition