TALLINN UNIVERSITY OF TECHNOLOGY
Faculty of Information Technology
Institute of Computer Science

Ilja Mašarov, 153004IAPM

DIGITAL CLOCK DRAWING TEST IMPLEMENTATION AND ANALYSIS

Master's thesis

Supervisors: Sven Nõmm, PhD
Prof. Aaro Toomela

Tallinn 2017

TALLINNA TEHNIKAÜLIKOOL

Infotehnoloogia teaduskond

Ilja Mašarov, 153004IAPM

KELLA JOONISTAMISE TESTI DIGITALISEERIMINE, IMPLEMENTATSIOON JA ANALÜÜS

Magistritöö

Juhendajad: Sven Nõmm, PhD

Prof. Aaro Toomela

Tallinn 2017

Author's declaration of originality

I hereby certify that I am the sole author of this thesis. All the used materials, references to the literature and the work of others have been referred to. This thesis has not been presented for examination anywhere else.

Author: Ilja Mašarov

08.05.2017

Abstract

The main focus of the present thesis is initial implementation of digital clock drawing test that could be used and would provide applicable results outside of the academic environment. Side goal is to gather the data and analyze the pilot study results to determine whether it is possible to find significant parameters that would make it possible to differentiate healthy and cognitively impaired individuals based on the extracted features and is it worth moving forward in this direction.

The present thesis has its footing on top the previous related works in the area and can also be considered as a first step towards measuring the role of motion mass parameters, pressure, azimuth and altitude angles of the pen in the diagnosis process - which were not evaluated in the previous research. Since the amount of received data is not enough to evaluate the role of those parameters for certain, this work can be considered as a pilot study.

In scope of the thesis, the initial implementation of digital clock drawing test was developed from scratch. The implementation includes part of the iPad application responsible for gathering the data, back-end services for receiving the data, parsing, transforming it, classifying the strokes and extracting the features from the drawing. The results were evaluated, discussed and analyzed.

The present thesis is written in English and is 78 pages long, including 5 chapters, 15 figures and 5 tables.

Annotatsioon

Kella joonistamise testi digitaliseerimine, implementatsioon ja analüüs

Käesoleva väitekirja peamine eesmärk on kella joonistus testi digitaliseerimine ja algusrakendamine, mida võib kasutada ja mis võib anda kohaldatavat tulemust akadeemilisest keskkonnast väljas. Lisaeesmärgiks on koguda andmeid ning analüüsida pilootuuringu tulemusi, et teha kindlaks, kas vaadeldud omadustele põhinedes on võimalik määratleda märkimisväärseid parameetreid, mille abil saaks eristada terveid inimesi kognitiivsete häiretega inimestest. Lisaks tehakse kindlaks, kas sellise testi suunas liikumine on mõistlik või mitte.

Käesolev väitekiri põhineb varasematel teemakohastel töödel ning seda võib pidada esimeseks sammuks pastapliiatsi liikumise massi parameetrite ning rõhu, asimuuti ja altituudi nurkade diagnoosi rakendamise suunas - eelmistes uuringutes neid ei hinnatud. Kuna ammutatud andmete hulk pole piisav, et hinnata nimetatud parameetrite rollide olulisust, võib pidada seda pilootuuringuks.

Esitletava väitekirja raames arendati digitaalse kella joonistamise test nullist. Testi rakendamise hulka kuulub ka iPadi rakendus, mille ülesanneteks on andmete kogumine, sõelumine, teisendamine, väljastamine ning tehtud liigutuste määratlemine ja joonisel ilmnevate omaduste eristamine. Saadud andmete baasil toimus hindamine, arutlemine ning analüüsimine.

Lõputöö on kirjutatud inglise keeles ning sisaldab teksti 78 leheküljel, 5 peatükki, 15 joonist, 5 tabelit.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Dementia is an expansive category of brain diseases that affect a person's daily functioning in a form of a decrease in the ability to think, remember, and perform daily activities. The most widely spread type of dementia is Alzheimer's disease, which is about 50-70% of the cases. Other types include vascular dementia, Lewy body dementia, frontotemporal dementia, Parkinson's disease and many others.

There is no cure for dementia as of today, but in many cases it is possible to dramatically slow down the development of cognitive impairment if the disease is caught early enough. Which means that constant screening and the possibility of early diagnosis are playing a very important role in the field.

Dementia is mostly prevalent among the elderly individuals, who experience natural decline in cognitive and physical abilities. Which means that the first signs of a disease can be easily overlooked even by the experienced clinicians, because they can be seen as part of the natural aging process.

Most popular means for dementia screening are manual pen and paper tests. The clinicians are asking patients to draw or write something on a paper, while observing the behaviour of drawing or writing process. Different manual scoring systems exists to evaluate the results of the screening tests, mostly based on the clinician's subjective assessment.

The main drawback of pen and paper tests is that subtle changes in patient behaviour could not be tracked or even noticed by the human eye. That's where

the technology comes to the rescue. There are numerous digital pens, tablet computers, touch screen smart phones and many other devices that makes it possible to record the drawings with very high accuracy and additional parameters such as time, pressure, angles of the pen and many more.

Numerous studies in the last decade suggest that using technology for recording the tests together with machine learning algorithms for screening them might be very beneficial for detecting subtlest changes in the patient behaviour. Which means that with the help of technology clinicians might be able to detect the dementia much, much earlier than in the current environment.

The huge amount of research papers and articles about the topic suggests that the idea of digitizing the dementia screening tests is quite popular nowadays in the academia, and many researches are working on improving accuracy and proving that the use of technology is definitely beneficial. But as of today, the technology is not being used anywhere, there are no digital versions of tests used or acceptable in the clinical facilities.

The focus of this thesis is to make first steps towards implementing the digital clock drawing test that could be ready to use outside of academic environment. The idea is to implement the test itself with the proper infrastructure not only for gathering the data and performing research, but for a real-life usage in the medical facilities.

The thesis is organized in the way of the implementation workflow. Introduction chapter describes the ideas and benefits of digital versions of common manual tests together with the details of clock drawing test. Next part of the introduction chapter explores related work and explains the motivation behind the thesis.

Second chapter goes into the implementation detail of the digital clock drawing test. The overall infrastructure is described, data acquisition part describes the way how the data is gathered and parsed. Stroke classification section explains how the strokes get classified using machine learning techniques together with other algorithms.

Third chapter goes into the analysis. First, the process and logic behind feature

extraction is explained together with descriptions of different types of features. Next, the implemented tools for the analysis of extracted features are described together with the methodology of statistical analysis, mean comparison of the features and measure of correlation coefficients between them.

Discussion chapter opens a discussion about the achieved results, the problems, possibilities and directions for further development and analysis of the digital clock drawing test. The conclusion chapter summarizes the work. Followed by the acknowledgments and bibliography sections.

1.1 Background

The "Clock Drawing Test" - a pencil and paper test, that has proven effective in helping to diagnose cognitive impairment that might indicate different neurological disorders such as Alzheimer's disease, Parkinson's disease, and other types of dementias. Initially, it was used only to assess visuo-constructive abilities, but over time it has become clear that abnormalities in a clock drawing also indicate other cognitive impairments.

The clock drawing test is often used as a part of commonplace regular screening for cognitive change, performed by neuropsychologists, neurologists and other types of physicians in the medical facilities. There are some varieties in the test, and the simplest type of clock drawing test asks the subject to draw a clock on a blank sheet of paper, showing certain time. There is no time limit for the test. The result of a clock drawing test might look like the one shown on Figure 1.1.

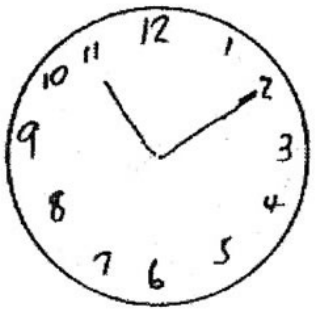

Figure 1.1: Example of Clock Drawing test

Clock drawing test can be performed in several different ways. The most popular type of clock drawing test is a free-drawn method, where the subject draws the clock entirely from memory. In case of other method, which has a pre-drawn circular contour of clock face, the subject is asked to draw only the numbers and clock hands using the existing contour. In many cases, the subject is asked to draw a clock hands pointing at a certain time, in other cases the clock hands are completely excluded from the test.

Clock drawing test has many different scoring systems. Those systems start from a very simplistic nominal scale with only right and wrong scores and ends with the

very detailed 31-point score systems. Almost all of the know scoring systems rely solely on the clinician's subjective judgment of the drawing.

The clock drawing test might be used for two purposes, one is diagnosis, another is screening. Screening - is a process to distinguishing the healthy individuals and cognitively impaired patients. The diagnosis - is a process of classification-distinguishing between the different cognitive illnesses. Or, more simply, the screening - is detection whether the patient is healthy or have a cognitive impairment, and diagnosis - is detection of which type of cognitive impairment the patient have.

1.2 Problem statement

In 2017, recent estimates suggest that nearly 44 million people worldwide have Alzheimer's or a related dementia, and more than 10 million people are living with the Parkinson's disease. The numbers are growing every day. As of today, there is yet no cure for cognitive impairments such as those, but it is possible to slow them down if caught early enough. Hence early detection makes a big difference.

One of the most popular tests that has been used for many years as a screening tool to differentiate healthy individuals from those with cognitive impairment is the "Clock Drawing Test" (CDT) - a pencil and paper test. In most of the cases the CDT is used by primary care physicians, neuropsychologists, neurologists as part of a general screening.

But there are major drawbacks in the manual use of the test. While there is a variety of manual scoring systems used by clinicians, these systems often rely on the clinician's subjective judgment of the drawing. Another drawback is that CDT cannot be delegated to the nurse and requires the presence and observation of a clinician during the whole drawing process, which might take up to 20 minutes per patient. Yet another disadvantage of manual scoring is the fact that human eye cannot register and take into account tiny details of the drawing, or measure the pressure of the strokes - those subtle details that might play a significant role in detecting the cognitive impairment during the early stages.

There has been several attempts to digitize the CDT, accompanied by the research studies that has proven the fact that machine learning techniques can achieve very high accuracy in screening for cognitive change, thus might help the clinicians to detect cognitive change at much earlier stage. But as of today, the technology is not being used in the hospitals. One of the possible reasons of this is the timing. Clinicians were not ready to trust the technology before, so there was no interest from their side, and the hardware was not good enough. Further, several gaps can be identified in the existing research:

- Pressure information was not used.

- Azimuth and Latitude angles of the pen were not used.
- The role of motion mass parameters is not measured enough.
- No use of deep learning algorithms in digit recognition.

Now, the computer-aided systems are starting to be more and more involved in the medicine. The new technology is being used progressively every day. Today, the clinicians are adopting and showing genuine interest in the newest technological developments. That's why the plan of the thesis project is to implement the digital Clock Drawing test in close collaboration with clinicians and trying to fill the gaps of the existing research in the process.

In the scope of a "Startup project" (ITX8549) course, together with a team we have started developing the common infrastructure for data acquisition and storage, which include the iPad Pro application and several back-end services. But this infrastructure was not sufficient and not yet ready for being used in the medical facilities.

In scope of this thesis work the whole infrastructure should be overhauled according to the feedback from clinicians, it should be more reliable and robust. It should be thoroughly tested and stress-tested. The scope of the thesis will try to include the implementation of the following items on top of the initial infrastructure:

- Data preparation (Means for getting the data from the patients, iPad application and infrastructure)
- Stroke classification. (Backend services that use machine learning and other methods to classify each stroke as part of the clock)
- Feature extraction (Using the results of classification for generating the features for analysis and machine learning models)
- The analysis and interpretation of the results. (Mean comparison between the features of impaired and control individuals)

1.3 Motivation

Machine learning is becoming an essential part of our lives. It it being used in almost any digital products that we currently use on a daily basis, be it a spam filter in the email client, face-recognition on social media, recommended products in an online-store and so on.

And it is becoming clear that using machine learning techniques is beneficial for people to use in many areas, especially in medicine. The main problem with health sector is the fact that it is quite conservative area when it comes to adopting new technology, but it is understandable, given the fact that people's lives are involved.

Another problem in medicine is data protection. Almost any personal record in medical facility has a lot of personally identifiable information, which should not get into the wrong hands and should be protected. The use of new an experimental technology is not welcome because it provides additional dangers in leaking or disclosing personal information.

The screening tools such as clock drawing test, and others might be the starting point to the gradual adoption of machine learning technology in medicine, because it is not something life-critical, it is simply an aid to the clinician that could help detect the disease earlier. And it does not rely on any personally identifiable information, it only relies on the data about the drawing which is completely anonymous. So there is no threat of accidentally disclosing the information. It's even the opposite, this anonymous information might or, even, should be shared and used in the further research without any damage to the patients.

Clock drawing test is one of the most popular fine motor tests used nowadays. It's easy to use, it's efficient and in most of the countries it is a mandatory part of cognitive change screening procedure. Which makes it a great starting point.

Even though there has been many successful studies and attempts to digitize the clock drawing test, as of today - it is not used in the hospitals. The data from the research is also not publicly available, which makes it impossible to continue the research or base new findings on the previous ones.

If there's even the slightest chance that the digital implementation of clock drawing test might help identify mild cognitive disease earlier than the clinician alone, the it is certainly something that should be implemented and used in the medical facilities.

The idea for this thesis has evolved from the real need, from the cooperation with one of the clinicians who have the real interest in using the digital clock drawing test in one of the Tallinn's hospitals. Which means that this work is not based solely on the assumptions about the need of this tool, but it's based on a genuine need and interest.

Having previous experience with implementing the digital version of Poppelreuter's overlapping figure test and conducting the research might also help me in the implementation of digital clock drawing test.

1.4 Related work

The relationship between ability to draw and cognitive impairments has been established a long time ago. In one of the studies [1] it was shown that patients with Alzheimer's disease draw images with broken perspective, spatial relationships between objects, simplification and overall impairment in comparison with the control subjects.

It has been shown [2] that the clock drawing test is seen as a complex multidimensional measure that should be used together with other cognitive tests in the neuropsychological screening. To some degree, the clock drawing test allows to measure the semantic knowledge, visuospatial abilities, executive functions, and general cognitive functions of a patient.

Several studies [3, 4] also suggest that qualitative analysis of a clock drawing could also demonstrate the profiles of distinct types of dementia and their differences. In general, the clock drawing test is useful distinguishing between Alzheimer's disease (AD), Frontotemporal Dementia and Vascular Dementia (VD), but shows limited value in differentiating between Alzheimer's disease, Dementia with Lewy Bodies (DLB), and Parkinson's Disease Dementia (PDD). The analysis of drawing errors has been shown [5] to have a greater impact than overall accuracy of a drawing in differentiation between different forms of dementia.

The CDT is not a perfect solution though, because sometimes [6] the errors and misplacement of digits on the clock drawing may be caused by the general disturbance in the conceptualization of time rather than cognitive impairment. That's why the test cannot substitute the physician, but should act as a tool and help.

Some of the studies suggest [7] that using only command clock (free-drawing) alone is not always effective to distinguish demented individuals from the control group. The reason is that drawing the command clock requires remembering instructions, verbal and visual memory, spatial planning, intact auditory comprehension and the ability to persist in drawing. It means that a low score of the command clock can be explained by other reasons rather than dementia. The copy clock, on the other hand has less demand on memory, but more demand for visuospatial

integration and inhibitory functions. Which means that in order to increase the effectiveness of the test, the results of both - command and copy tests should be compared and evaluated together.

The free-drawing version of clock test is also proven [8] to be more cognitively demanding and more accurate in detecting mild/early cognitive impairment. The Mild Cognitive Impairment (MCI) is frequently described [9] as a transition between normal aging and Alzheimer's disease, so it can be particularly interesting and challenging to detect it as early as possible.

Several studies [10] have shown that clock drawing test is reliable, but in a certain rare circumstances it does not provide a valid results. It happens in cases, when screening is performed with the patients that have four or less years of formal education. This is something that should be kept in mind. It has also been found [11] that people with higher levels of education were performing much better and doing fewer errors on a different scales, overall, the errors on a clock test are more common among older than younger individuals.

One of the clock drawing test studies [12] suggest that the analysis should be focused more on the drawing errors rather than overall drawing accuracy. Researchers suggest that in many cases it is enough to assess only six errors to get a good results, which include: missing numbers, number substitutions and repetitions, inaccurate time setting, missing clock hands and sometimes even refusal to attempt clock drawing. As we can see, special attention should be paid to the error detection during the clock test.

As of today, several manual scoring systems, such as mini-cog [13], clox [14], Mendez's [15], Shulman's [16] and others [17, 18] are being used to evaluate the results of the clock drawing test in the hospitals. But it has also been shown [19], that manual scoring systems has no significant differences between each other and that all of them provide only modest results (area under the curve from 0.60 to 0.72).

Different manual scoring systems were compared multiple times [20, 21], and the studies indicate that one of the best scoring systems correctly identify around 85% of

subjects, whereas on average, manual scoring systems correctly identify around 70% of mildly demented individuals. The result is not very good, but as a comparison, in case of physicians personal judgment based on a medical record's from patient history, they were able to identify only 24% of mildly demented patients.

Another problem with manual scoring is that manual scoring systems often rely on the clinician's subjective judgment of properties of the drawing. Some research suggest [22] that combining the existing scoring systems together with machine learning methods might allow significantly better screening of cognitive conditions, remove subjectivity and maintain some level of interpretability of the model.

The hand movement and it's relationship to Parkinson's disease has been modeled and measured not only in a clock-drawing test. Another popular test is a spiral test, where patients are asked to draw a spiral. In one of the studies [23] researches have created and analyzed the results of a digital version of a spiral test, but the research is more focused on creating the mathematical model that can be used for spiral drawing. In more recent study [24] the Luria'a alternating series test was also digitalized and the role of kinematic features was measured and analyzed. Another study [25] was focused on the recognition and analysis of contours drawn during the digital version of the Poppelreuter's overlapping figures test.

The digital version of a clock drawing test makes it possible to notice and measure extremely subtle behaviours, such as brief pauses, accidental pen strokes - which might play a huge role in a detection of mild cognitive impairment. This subtle behaviour was also investigated in an attempt [26] to analyze the clock drawing test performed by different aging group patients with a depression, where it has been shown that the digital clock drawing test differentiated the aspects of psycho-motor slowing in a depression, regardless of patient age.

There are several approaches to create a digital version of a clock drawing test. One approach is to use a digital pen together with a non-digital paper, and another approach is to use a digital tablet together with a digital pen. One of the first attempts to create a digital clock drawing test on a digital tablet [27] was more focused on the interface design, trying to implement it in a way that is usable by both patients and clinicians. It makes a very basic use of the timing information,

does simplistic digit recognition and a little bit of analysis on a patient performance. It does not, however, implement any automatic detection of cognitive conditions and does not describe any complexities involved in the process. This was more a proof of concept about the possibility to perform the test on the digital tablet.

One of the first attempts to use state-of-the-art machine learning methods in the digital clock drawing test [28] has described the use and effectiveness of a large collection of machine learning algorithms, including SVMs, boosted decision trees and random forests. The SVMs has been shown to produce the best results in the classification of impaired and healthy patients. This research also describes the use of a different data mining algorithms, such as FPGrowth, Apriory and Bayesian List Machines (BLM) in an attempt to create decision models with high accuracy and high comprehensibility at the same time. The researchers also argue that the current practice in cognitive assessment which assumes that average scores indicate the absence of cognitive impairment is questionable, because patients often hide early and subtle impairments by thinking harder or working longer in a ways that are not noticeable by the clinician.

Most of the studies about digital clock drawing test only describe the process and techniques for screening and diagnosis parts of the test, but only very few describe the process of stroke classification, which is an essential part of the test. It is really important to recognize and classify each stroke of the drawing and it's belonging to one of the clock elements, such as clock-face, digits, arrows, in order to generate proper features. In one of the studies [29] researches describe the techniques they have used for classification and the prevailing problems during this process. Although, they mostly describe the techniques they have used for arrow and clock-face recognition, because for digit recognition they have used an existing solution, which is the "Tablet PC Platform SDK" from the Microsoft. But again, as with most of the studies, the software used in this study is more suitable in case of using it in academic environment, rather than real medical environment. So there is a clear gap between the possibility to analyze the data "offline" versus the "live" analysis which is being performed immediately after the test has been completed by the patient. None of the studies were found that would address this problem or describe

the tools that will allow this system to work as a back-end service for the test.

The first attempt [30] to tackle the trade-off between the accuracy of prediction and interpretability of the results by the clinician is comparing the black box off-the-shelf machine learning techniques, such as Gaussian SVM, random forests, CART, C4.5, boosted decision trees, regularized logistic regression with a high-interpretability scoring techniques such as Bayesian Rule Lists (BRL) and Super-sparse Linear Integer Models (SLIM). The researchers have tried to build classifiers that are are more accurate, but as interpretable as existing scoring systems and compare them to more traditional machine learning techniques. As expected, the black box techniques show the best accuracy in both, screening and diagnosis tasks, compared to interpretable machine learning techniques, but both of the techniques are better than manual scoring performed by the clinician.

1.5 Linked studies

The present thesis has its footing on numerous related studies conducted within our department. The figure 1.2 is showing a simplified diagram of those related works. The blue rectangles show the completed works, the green rectangles indicate the ones that are currently in progress and the orange rectangle indicate the present thesis in relation to other works.

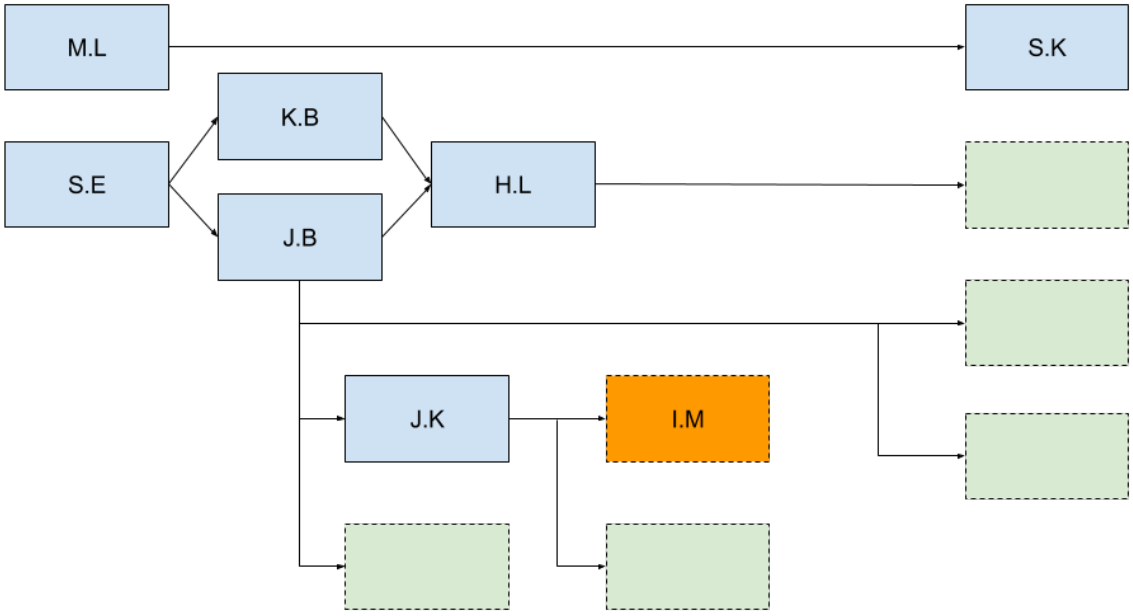

Figure 1.2: Linked studies

The rectangle with the M.L initials stands for one of the first related works, which was a Master's thesis "Application for Gesture Based Control of the "Pioneer" Robot with Manipulator" [31] authored by Mihhail Lapuškin in 2012. It was one of the first works in our department related to human motion motorics and human-machine interface.

This work led directly to the more recent recent Bechalor's thesis "Recognition of Hand Gestures Using Bezier Curve and K-nearest Neighbors Method" [32] authored by Siim Kirme in 2016. Which is represented by the rectangle with initials "S.K". This work continues the research in the area of gesture recognition. Which is related to the studies of gross and fine human motor functions.

One of the first works in our department related to data acquisition and statistical

analysis started from the Master's thesis "Gesture Based PC Interface with Kinect Sensor" [33] authored by Samet Erap in 2012. This work was a kick-start for later studies related to motor function data acquisition. This work is marked with the rectangle "S.E".

Next work in this area was a Master's thesis "Monitoring of the human motor functions rehabilitation by neural networks based system with Kinect sensor" [34] authored by Kirill Buhhalko in 2013 and represented by the rectangle with initials "K.B". Which uses the Kinect sensor for automatic monitoring and supervision of human motor functions.

The rectangle with initials "J.B" represents the Master's thesis "Alternative Approach to Model Changes of Human Motor Functions" [35] authored by Jevgeni Boruško in 2014. Which was focused on the gross motor function modeling for measuring the progress of rehabilitation or cognition using the performed motions.

The rectangle with initials "H.L" stands for the Master's thesis "Multi-Kinect system for acquisition of turning motion" [36] authored by Helena Lissenko in 2015. Which was focused on measuring and evaluating the turning motion using multiple Kinect sensors.

The rectangle with the "J.K" initials is representing the Master's thesis with the name "Quantitative Analysis of the Kinematic Features for the Luria's Alternating Series Test" [37] authored by Julia Koženkina in 2016. This is really important research in scope of present thesis, because it can be called the foundation of present work.

This study has evaluated the role of kinematic features in the digital version of Luria's alternating series test and has proven that there is a significance in those features that would allow the differentiation of healthy controls and cognitively impaired individuals.

After this study, together with the author we have co-authored and published the research paper "Recognition and Analysis of the Contours Drawn during the Poppelreuter's Test" [25], where we have used the same platform to implement the digital version of Poppelreuter's overlapping figures test. The experience of

conducting this study should help me in the implementation of digital clock drawing test.

All of the green rectangles represent the studies which are based on the works described above and which are currently in progress. The amount of ongoing research indicates that there is a lot of interest in the area. As it is clear from the descriptions there are several directions of the studies and different technologies, but almost all of them are related to measuring the human motor functions, be it a gross motor or fine motor functions.

Described works provide a common knowledge and a great foundation for implementing the digital clock drawing test. The current thesis will also possibly become the turning point in the related studies, because it is the first work which used the cloud based solutions for data acquisition and storage. All of the previous works rely on the "offline" data gathering, which is not convenient and does not allow providing results in the real time.

Chapter 2

Implementation

2.1 Overview

The whole project consists from several phases. And each phase consists of multiple parts. The simplified overview of the development phases can be described as follows:

1. Data acquisition
    (a) iPad application development
    (b) Back-end infrastructure development
    (c) Testing
    (d) Getting the data from the patients
2. Feature extraction
    (a) Stroke classification
        i. Data transformation
        ii. Circle classification
        iii. Clock hands classification
        iv. Digit classification
            A. Deep convolutional neural network model training
        v. Other elements classification
    (b) Feature extraction

3. Training/Analysis
    (a) Statistical analysis
    (b) Creating machine learning models (proof of concept)
4. Deployment
    (a) Extracting the code, deploying the models

During the first phase (Data acquisition), the most important goal is to prepare the application and infrastructure for the data acquisition process. At the end of this phase, the iPad with installed application will be given to the doctors, who will organize the meetings with the patients to perform the tests.

The goal of the first phase is to collect the real-world data from the impaired patients and the control groups. At the end of this phase it's important to have working and well-tested application and infrastructure for recording and storing the acquired data.

The second phase (Feature extraction) is one of the hardest and most important phases of the project. After acquiring the data, first, it should be transformed from the raw format to the format which will be appropriate for the later use.

After data is transformed, each stroke should be classified as one of the clock elements. For this classification several approaches are being used: the heuristics approach and the machine learning approach. The heuristics approach uses the different heuristics during the classification, such as the fact that in most of the cases, the first stroke is a circle or the fact that in majority of the cases, the circle stroke is the biggest stroke in the drawing, and so on.

The machine learning approach is using the convolutional neural networks for digit recognition, k-means clustering for detecting the groups of digits, and so on. After each stroke is being classified, the features are extracted. The end result of the second phase is a collection of algorithms to classify each stroke of the drawing, to extract the features and present them in the format of a single row in the dataset.

The third phase (Training/Analysis) of the project is statistical analysis of the features, extracted from the acquired data. After separating the data from control

group and data from cognitively impaired patients and extracting features from both datasets, the mean comparison between those sets of features should be performed.

Each feature is extracted from both datasets in isolation and creates a vector of values. In the result, we will have two vectors for each feature, one vector will have sample of values from control patients, another vector will have sample of values from cognitively impaired patients. Then both vectors are compared and p-value is measured.

After p-values are measured, the results are sorted and analyzed. If any features have significantly small p-value (less than 0.05), then this feature is considered to be potentially significant in differentiating the healthy and unhealthy individuals.

Second part of the third phase is actually using the two datasets to train machine learning models that would try to predict whether the patient is healthy or cognitively impaired. But this step requires large amount of data to present any kind of results that could be accepted. So this step is more a proof of concept.

The main goal of the final phase (Deployment) is to transform all the code used in the previous sections into the format, which will be appropriate to use in the production environment. During the first three phases of the project the Jupyter notebooks were used as a single tool to perform all of the calculations.

The Jupyter notebooks make the work much easier during the initial phases, because it allows to execute the blocks of code independently, it also allows to insert the markdown text between the blocks to describe the code or intention. Another useful feature is a support of an inline plotting and console output.

But the problem with the notebook is that the code cannot be reused between different notes. Another problem is that since the majority of code is written in the interactive fashion, it is much harder to port the code to make it usable on the web server. So during the fourth phase, all of the code should be extracted into separate Python modules, which will make it possible to reuse it in different notebooks, and will allow to use the same code in the web servers that will run in production environment.

2.2 Tools

The present thesis consists of multiple separate projects. Each project used **Git** for version control and **Bitbucket** as a cloud for git repositories. Initially, the **GitLab** was used as a cloud for git repositories, but during the thesis development, this company had big incident when the whole production database was lost together with all data. After this incident is was decided to move to bitbucket.

Trello was used for organizing the tasks and keeping the track of everything that should be done. **ShareLaTex** was used as a main tool for creating and maintaining the written part of this thesis. The thesis was written using the **LaTeX** document preparation system.

The development of iPad application was done using **Swift** programming language and **Xcode** as main integrated development environment (IDE). Sometimes the **AppCode** was used as an IDE, because it makes refactoring and code navigation much more pleasant, than Xcode. Several iPad Pro devices were used for development and testing.

All of the back-end development was done using **Python** programming language. Stroke classification development was initially done in **Jupyter** notebooks, because it allows to create prototypes faster, show plots and write descriptions very conveniently. But during the later stages of development, the code was moved from notebooks to Python modules, the **PyCharm** IDE was used.

The development of deep convolutional networks was done using the open-source library **TensorFlow**. It was chosen becuase it's the newest, highly optimized and very popular library developed mostly by the specialists from Google. The popularity of the library and the size of the community makes it easier to find tutorials or documentation.

Backend services were built using the **Flask** Python web framework. It was chosen because it's very lightweight and easy to get started with. **Boto 3** Python library was used to integrate web services with **Amazon AWS** infrastructure. The backend services were deployed to the **AWS EC2** (Elastic Compute Cloud) in-

stances using the **AWS Elastic Beanstalk** orchestration service. The data from the tests was saved to the **S3** (Amazon Simple Storage Service) buckets.

Since the main development language was Python, there are numerous Python libraries being used. Here i will mention only the most notable of them. For plotting the charts and plots, **Matplotlib** library was being used. For most of the computations and some data structures, the **NumPy** library was used together with **Pandas**. For statistical analysis, the **SciPy** library was used. And finally, for machine learning algorithms the **scikit-learn** open-source library was used.

2.3 Infrastructure

The initial implementation of infrastructure was started in scope of the "Startup Project" (ITX8549) course we had in the previous semester. The whole infrastructure consists of the iPad application and several separate back-end services that communicate with each other, the simplified diagram is displayed on the figure 2.1. But this initial infrastructure was not sufficient for the needs of the clinicians, and in scope of thesis work it was greatly overhauled and improved according to received feedback and new requirements.

Among the digital clock drawing test logic, the infrastructure also include error recognition API and descriptive statistics API, all of which work in parallel.

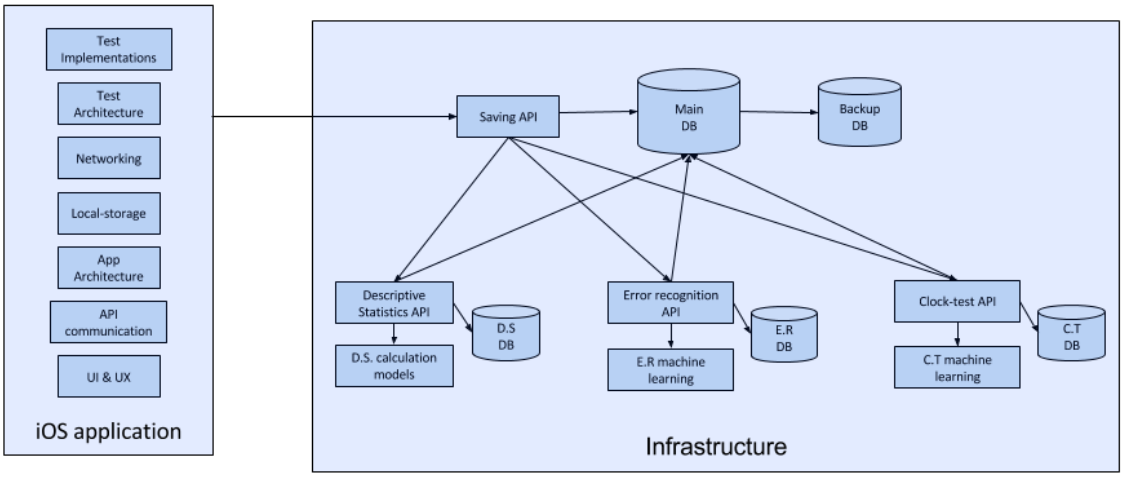

Figure 2.1: Initial infrastructure

During the process of drawing, the iPad records 240 points per second, where each point has several values. Since we have so much data being recorded per second, it is possible that the single test data in the JSON format might be more than 10 Megabytes - which means that the special measures and preparation must be implemented in the saving API to avoid data loss.

Initially, in our infrastructure we were using MongoDB for storing the data, thus implementing the backup mechanism and managing the database ourselves. But in scope of the thesis work - it was decided to migrate data storage to to Amazon S3 bucket, which allows us to delegate all of the reliability and backup maintenance to the experts in Amazon, instead of reinventing the wheel and implementing ev-

erything by ourselves. The AWS cloud infrastructure is one of the most reliable infrastructures available in the Internet.

Initially, the saving API was implemented in the GO language, but in scope of the thesis work it was decided to make it more consistent and use Flask framework (Python) to make it easier and more natural to use popular Python libraries.

In scope of thesis work, the saving API was also dramatically simplified - it doesn't even try to parse the JSON data - it simply takes all of the incoming traffic, and saves it into JSON file in the S3 bucket. This simple approach guarantees that even if due to some reason the saving API will receive malformed JSON or partial data - it is still going to be saved. Avoiding data loss was one of our main priorities.

Main parts of our infrastructure are:

- iPad application - the application (written in Swift) for performing the screening tests and sending the acquired data to back-end services
- Amazon Elastic Compute Cloud (EC2) - instances of virtual machines where we run our services (e.g. Saving API, Digit recognition API, etc.)
- Amazon Simple Storage Service (S3) - object storage with very high durability and user interface to manage the data. All of the test data is saved in the S3 bucket.
- AWS Elastic Beanstalk - orchestration service that makes the deployment and management of our services (EC2 machines) easier and faster.

2.4 Data acquisition

During the digital clock drawing test process, two individuals are involved, one is a patient and another acts as an instructor, who could be the doctor, nurse or any other employee of medical facility.

The iPad Pro together with an Apple Pencil are provided to the patient, then he is given short instructions about the tests and he may ask any questions about them. The instructor launches our application, enters the patient identification number, which is going to be used afterwards to match the data we have from the iPad application with the patient data provided to us by the clinicians offline.

After entering the patient identification number, the screen with the test selection is displayed, as shown on figure 2.2. The clinicians might configure this screen to show only certain tests. This screen also shows the progress of the same patient session to make it easier to track completed tests.

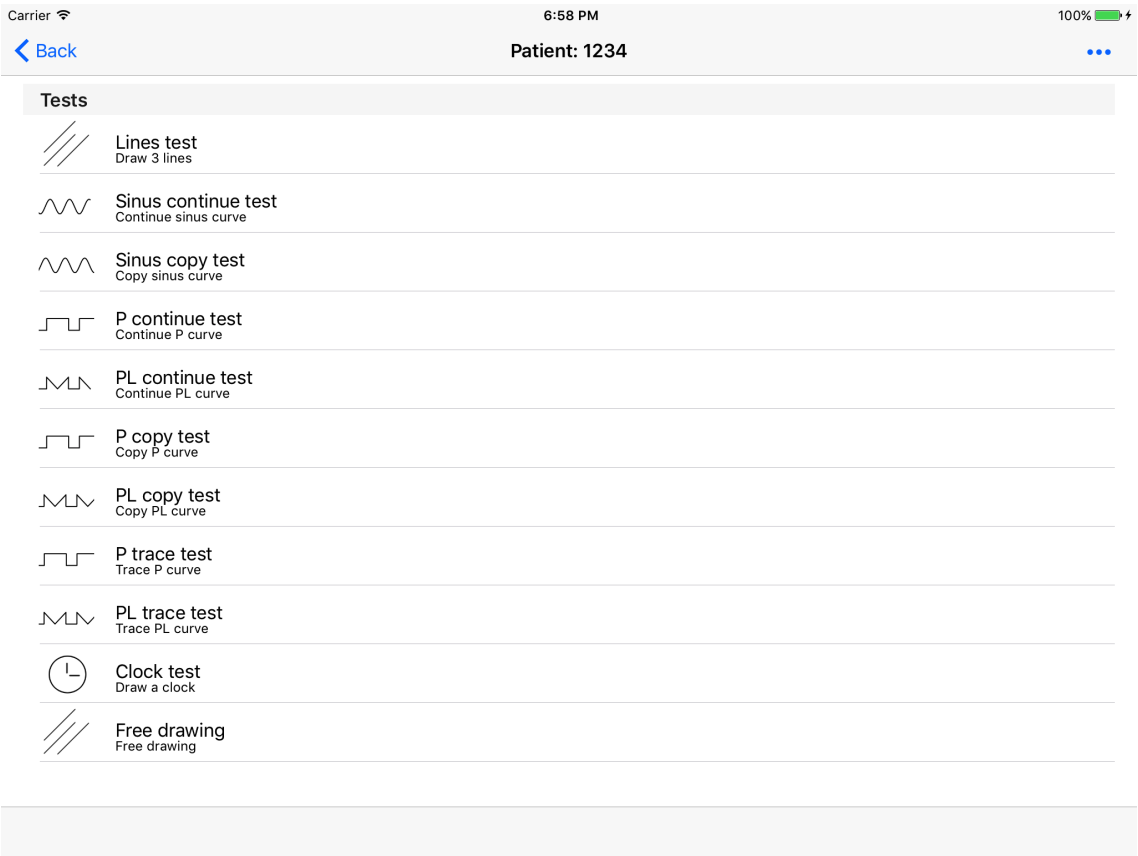

Figure 2.2: Screenshot of iPad application

After selecting the clock test from the list of tests, or after completing the previous test and clicking the button "next", the clock drawing test screen is opened. This is simple empty screen with the buttons "back" and "next" on the status bar on the top of the screen. The empty area inside of the screen is for drawing.

Since the clock drawing test is one of the many tests performed during the medical screening not much of the attention is spent on it in particular. The patient is simply asked to draw a clock, he is not asked to draw the exact time.

After the test has been completed and the instructor will trigger the next one, the iPad will check whether the Internet is available and if it is, will convert the drawing data into JSON format and send the it directly to our servers. If the Internet is not available at the moment or if sending the data has failed, the data will be saved in the local storage of the iPad.

Every time the application is opened - it will check whether the Internet is available and if there are any unsynchronised tests, and will try to send them asynchronously in the background, hidden from the person using the application. If because of some reason sending data is not an option, it is possible to extract the data using the XCode application with developer's provisional profile - which ensures that only the developers of the application has access to data.

2.5 Stroke classification

The classification phase is one of the hardest phases in the whole project. During this phase the goal is to recognize each stroke. Which strike belongs to the circle around the clock? Which strokes belong to digit 6, 5? Which strokes belongs to the clock hands, etc.

The problems arise when we start to think about all the details that could act as a hamstring in our classification problem. People draw digits differently, most of the digits can be drawn with the single or multiple strokes, there could be other marks or repeated digits. There could be crossed-out elements or incorrect digits. Their placement could be completely incorrect. There are many so-called edge cases that we need to consider.

In this phase the following main elements of the clock should be recognized:

- Clock circle (The contour of the clock face)
- Digits (Digits from 1 to 12)
- Clock hands (Arrows in the center of clock)
- Other elements (Marks that are not related to any of the above elements)

For each of the elements multiple approaches should be used. For circle recognition, the geometrical properties of the stroke should be taken into account together with different heuristics like the fact that the circle should usually be the longest stroke (considering that it will be drawn as a single stroke). The arrows could also be recognized by the combination of geometrical properties together with heuristics. The digits are the hardest part. For digit classification it was decided to use convolutional neural networks (TensorFlow) trained on the MNIST dataset.

But before classifying the digit, there is a lot of preparation work. First problem is that since most of the digits can be drawn with the multiple strokes, there should be a mechanism to find the strokes that belong to the single digits. The combination of different factors should be taken into account - the placement of the strokes, total number of strokes, the closeness and overlapping of strokes.

Different combinations of possible strokes that belong to a single digit could be validated with the trained models to see which combination yields the higher probability of it being a single digit. Next challenge is to transform the digit into appropriate format for using it with a model trained on MNIST dataset. The MNIST dataset consists of 55000 28x28 pixel images, which are being transformed into 28x28 matrices. In order to classify the strokes, they should be transformed accordingly, to match the MNIST format.

2.5.1 Data parsing

The data from the iOS application comes in the JSON format, which has array of strokes, and meta information, such as the patient identifier, session identifier, etc. Each stroke consists of array of points. And each point consists of the parameters, presented in a table 2.1.

Feature	Description
X-axis coordinate	X-axis in a Cartesian coordinate system
Y-axis coordinate	Y-axis in a Cartesian coordinate system
Timestamp	The interval between the now and 00:00 UTC on 01.01.2001
Pressure	Force of the touch, where 1.0 is the force of an average touch
Altitude angle	The altitude of the stylus, in radians
Azimuth angle	Angle of direction in which the stylus is pointing

Table 2.1: Point parameters

After the data in JSON format is loaded into memory, it is iterated and loaded into appropriate python classes, such as Point, Stroke, Drawing, etc. During the initial parsing, some of the features are already being calculated to make further calculations easier, those features include maximum and minimum x and y points in each stroke, start and end time of each stroke, length, area of each stroke, height

and width of the stroke, etc. The example of plotted RAW data is shown on the figure 2.3.

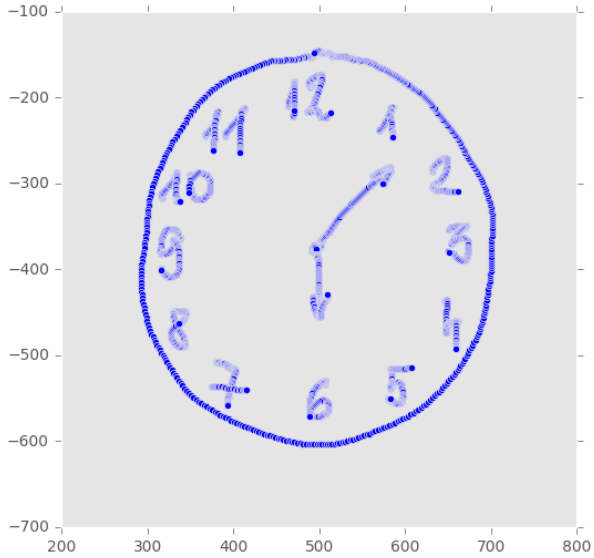

Figure 2.3: Raw data of the drawing from the iPad

2.5.2 Circle classification and analysis

Circle detection is based on several heuristics, or we could call them assumptions. First assumption is that the whole circle is probably drawn as a single stroke, because it's quite rare when someone start drawing circle that consists of several smaller strokes, so it is considered as a corner case. The second assumption is that in the majority of cases the circle stroke is the first stroke of the drawing, because most of the people are going to use it as an anchor and a boundary for the further strokes.

Third heuristic is that the circle stroke most probably will have the biggest length (in case if it is drawn as a single stroke), and will have the biggest covering area. The last assumption is that in majority of cases, the circle will be the outermost stroke of the drawing, meaning that it will be the closest stroke to the borders of the drawing screen.

During the circle classification process, each stroke is being tested against all of the described heuristics, where each heuristic has it's own weight, and according to that weight the probability of the stroke being a circle is increased in case if the heuristic holds true. If probability of a single stroke being the circle is more

than 80% and there are no other strokes with high probability, then this stroke is considered to be the circle.

As with many heuristics, most of the time they will hold true, but there are several problem cases when it will be impossible to classify the circle based on the heuristics. The most problematic is the case when the circle is drawn with multiple strokes, because this case ruins most of the assumptions. In this case none of the strokes will have a high probability of being a circle, which means that the next attempt to detect the circle will happen during the further stages of the classification - during the analysis of possible multi-stroke elements.

In case if most of the heuristics held true and the circle was classified, then the most important circle features must be extracted. First, the center of the circle should be identified, it is done based on the total width and height of the stroke, divided by two. After the center is identified, there are three "perfect" circles to be calculated for the further comparison and feature extraction:

- Inner circle
- Outer circle
- Mean circle

The "perfect" circles are required to measure the oblateness and ellipticity of the circle. The inner circle is calculated by taking the smallest (minimum) of width and height and using it as a diameter. The outer circle is calculated in the opposite way - by taking the largest (maximum) of width and height and using it as a diameter. The mean circle is calculated by using the mean euclidean distance (equation 2.1) from the center of the circle to the each point of the stroke.

$$d = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2} \tag{2.1}$$

The Figure 2.4 illustrates the original stroke (black), the center of the circle (red dot), the inner "perfect" circle (green), the outer "perfect" circle (magenta) and the mean "perfect" circle (red).

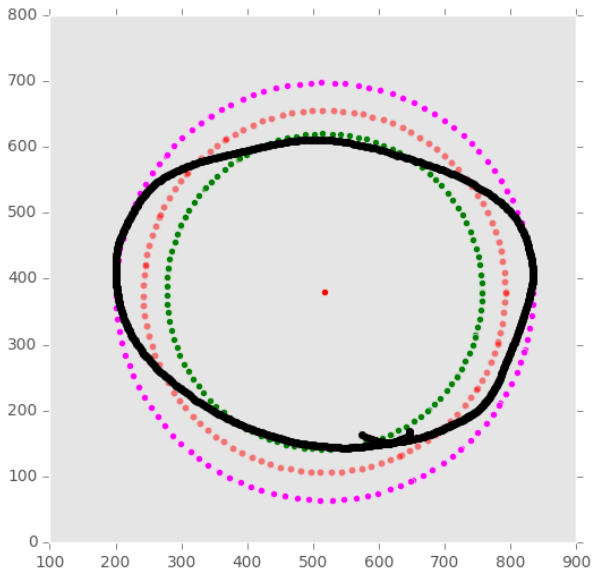

Figure 2.4: "Perfect" circles around the original stroke

Next step is to represent the roundness of the circle by the several features. First, for each of the "perfect" circles we are going to measure the mean squared error, shown on the equation 2.2 and the mean absolute percentage error, shown on the equation 2.3 between the "perfect" circle and the real stroke.

$$\mathrm{MSE} = \frac{1}{\mathrm{n}} \sum_{\mathrm{t}=1}^{\mathrm{n}} \mathrm{e}_{\mathrm{t}}^2 \tag{2.2}$$

$$\mathrm{M} = \frac{100\%}{\mathrm{n}} \sum_{\mathrm{t}=1}^{\mathrm{n}} \left| \frac{\mathrm{e}_{\mathrm{t}}}{\mathrm{y}_{\mathrm{t}}} \right| \tag{2.3}$$

After initial features of roundness are calculated, it's time to measuere the roundness using the internationally defined methods [38], which include the Least Squares Circle (LSC), Minimum Circumscribed Circle (MCC), Maximum Inscribed Circle (MIC) and Minimum Zone Circle (MZC).

Least square circle (LSC) - is a circle that separates the original stroke by separating the sum of total areas of the inside and outside of it in equal amounts. The error then can be measured as the difference between the maximum and minimum distance from the reference circle, which is shown on the equation 2.4.

$$F(x_c, y_c, R_c) = \min(\sum_{i=1}^{n} R_i^2) \tag{2.4}$$

Minimum Zone circle (MZC) - two circles (outer & inner "perfect" circles) are used as a reference for measuring the roundness error. The roundness error here is the difference between the radius of the two circles, which is shown on the equation 2.5.

$$F(x_c, y_c) = \min\{\max(\sqrt{(x_i - x_c)^2 + (y_i - y_c)^2}) - \min(\sqrt{(x_i - x_c)^2 + (y_i - y_c)^2})\} \tag{2.5}$$

Minimum circumscribed circle (MCC) - is defined as the smallest possible circle which encloses the whole stroke. Here the error is the largest deviation from the outer "perfect" circle, which is shown on the equation 2.6.

$$F(x_c, y_c) = \min\{\max[R_i]\} \tag{2.6}$$

Maximum inscribed circle (MIC) - is defined as the largest possible circle that can be inscribed inside stroke. The roundness error here is the maximum deviation from the inner "perfect" circle, which is shown on the equation 2.7.

$$F(x_c, y_c) = \max\{\min[R_i]\} \tag{2.7}$$

After calculating the features related to roundness of the circle, some additional features are calculated. This include the starting and finishing quadrants that represent the relative to the circle position where the stroke has been started and has finished.

The quadrants are identified in the relation to the central point of the stroke. The number of crossed quadrants is also measured as a feature, in most of the cases the number is four, but it's interesting to observe the cases when the circle stroke was drawn several times, or has an unusually long tail that crosses additional quadrant.

The direction of the circle is also identified, which could be either clock-wise or counter-clock-wise.

2.5.3 Clock hands classification

After classifying the circle, next step is to classify the clock hands. The classification is mostly based on an assumption that the clock hands are usually the most central elements of the clock, which holds in the majority of cases.

During the clock hand classification, all of the strokes are getting iterated and their distance to the central point of the circle is measured, since the circle was already classified and it's central point is known. After measuring the distances between all the strokes and the central point, and assuming that in majority of cases the standard deviation (equation 2.8) of the distances across digits should be relatively low, because the digits should be drawn around the circle, thus digits themselves form a circle with the radius of a distance from the central point. The clock hands are exceptions in that case, thus they make standard deviation in the distances much higher.

$$\sigma = \sqrt{\frac{1}{N-1}\sum_{i=1}^{N}(x_i - \overline{x})^2} \tag{2.8}$$

Using this knowledge we iterate over all of the strokes in the order of their closeness to the center and eliminate them one by one and measure the changes in standard deviation compared with the previous value. If eliminating the stroke decreases the standard deviation more than 5% - this stroke is considered to be related to the clock hands. Once the standard deviation stops increasing, the iteration stops.

2.5.4 Digit classification preparation

Preparation

For digit classification it was decided to use deep convolutional neural networks [39] trained on the MNIST [40] dataset using the TensorFlow [41] software library.

MNIST dataset is a freely available database of images of handwritten digits. MNIST dataset consists of 60000 training examples and 10000 testing examples. Each image is 28x28 pixels in size, from which the digit itself is 20x20 pixels. The figure 2.5 is showing one sample from the MNIST dataset.

Figure 2.5: Example MNIST digit

Creating the model starts with preparing the MNIST dataset for training. The training data in a dataset being used is a .csv file where each column represent a pixel (with total of 785 columns, last column represent a label) and each row is a training image (with total of 60000 images). Since each pixel might have a value ranging from 0 to 255, it needs to be converted to fit the range between 0 and 1. After conversion, the labels are extracted from the data and stored separately.

Next step is to transform the labels into "one-hot encoding", because in this format they work better with classification algorithms. The example of one-hot encoding is the following: we take the label of digit "3" and knowing that the total number of classes (labels) is 10, we will transform it into: [0, 0, 0, 1, 0, 0, 0, 0, 0, 0], where all of the other labels except the fourth (count starts with zero) are zeros.

The dataset is split into two parts, 50000 images are used for training and 10000 are left for validation. Validation data is used to evaluate the trained model.When the data is ready for training - it's time to configure the TensorFlow. TensorFlow

is an open-source machine learning library, developed by Google. In TensorFlow, the main unit of data is called the tensor. Basically, it can be thought as an array of primitive values with any number of dimensions. TensorFlow also uses a computational graph to perform the calculations.

Since the TensorFlow does very heavy calculations, the graph is needed to optimize the performance and run those calculations in a separate process. The computational graph consists of nodes, where each node is an operation. To evaluate the nodes of a computational graph, we need to run it within a session. Another important terms in TensorFlow are placeholders and variables. A placeholder is a promise that the value will be provided later. And the variable is a construct that makes possible adding the trainable parameters to the computational graph.

First, a placeholders for our images and for it's labels are created. Next step is to create a first convolutional layer. We need to define a variables for weight and bias. They are assigned with initial values, in case of bias it's a constant 0.1 and in case of weights, we create a tensor with a shape [5, 5, 1, 32] that consists the values with the normal distribution where the mean equals to zero and standard deviation equals to 0.1. As a good practice, weights should be initialised with some noise. This is done for the symmetry breaking, and to prevent zero gradients.

Neural Network structure

It is convention to apply a nonlinear layer after each convolutional layer. The purpose of this layer is to introduce some nonlinearity to the system that previously has been computing only linear operations. Some time ago, nonlinear functions like sigmoid were used for such purposes, but it has been found [42] that rectified linear unit (ReLU) layers work much better because they allow the network to train significantly faster without sacrificing accuracy.

Since on the first layer we are going to use rectified linear unit (ReLU) neurons, the ones that contain rectifier function shown on equation 2.9, it is also recommended to initialize the weights with a positive initial bias. It is done in order to avoid what is called "dead neurons".

$$f(x) = \max(0, x) \tag{2.9}$$

Convolution layer is generally used to get the main features of the data. In our case of digit classification the main feature is a shape of each digit. In a convolutional layer, 2 main parameters can be used to control the behavior of the layer. Those are the stride and the padding. Stride is how the filter convolves around the input volume. Padding is used us to control the spatial size of the output volumes. In our first layer, we are going to use stride with value 1 and zero-padding that will pad the input with zeros around its border.

Next layer is a pooling layer. Pooling is mostly used for downsampling of the data. The main purpose of pooling is to reduce the spatial dimension of the input, which will reduce the amount of parameters and weights by around 75% (which will dramatically decreases the computational cost) and will allow us to control overfitting. We are going to use 2x2 max-pooling, that will split the image into 2-pixel blocks and will only keep the maximum value for each block.

What is fascinating about neural networks - is that any nearal network can be used as a single layer in a multilayer neural network, in which case the output of one neural network can be used as an input for another. This approach allows us to create very big and complex neural networks with multiple layers - that's why it's called Deep Neural Networks. In our case we are going to use two convolutional layers with the pooling layer between them. Then densely connected layer which will be followed by the dropout layer. The readout layer is going to be our last layer. The simplified diagram of the convolutional neural network being used is shown on the figure 2.6.

The first convolutional layer will compute 32 features for each 5x5 patch. It will initialize the weight tensor with a shape of [5, 5, 1, 32]. The first two dimensions represent the patch size, next dimension is the amount of input channels, and the last dimension is the amount of output channels. To apply this layer, first, we reshape the input to a 4 dimensional tensor, where first dimension is the total number of images, second is image width, third is image height and the fourth dimension is

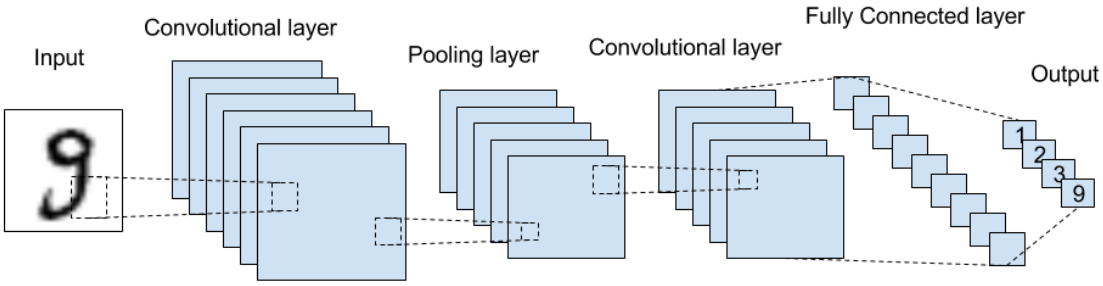

Figure 2.6: Simplified structure of convolutional neural network

the number of color channels - which is 1 in our case, because images are grayscale. After this convolution, the pooling layer will reduce the size of the output to 14x14 matrix.

The second convolutional layer has 64 features for each 5x5 patch. The weight tensor has a [5, 5, 32, 64] shape, where first and second dimensions are the patch size, the third dimension is the amount of input channels and the fourth dimension is the amount of output channels. The number of input channels is 32, because that's what we get from the previous convolutional layer. And since the input was down-sampled by the pooling layer, the second convolutional layer will pick up more general characteristics of the images.

Next step is to create a densely connected layer, or as it's also called - the fully connected layer. This layer is used for the high-level reasoning in the neural network. In a fully conneted layer, the neurons have full connections with all activations from the previous layer, which means that their activations can be computed with a simple matrix multiplication and a bias offset afterwards. In our case since by now the image size is reduced to 7x7, our fully connected layer will have 1024 neurons.

After densely connected layer, we create a dropout layer. It has been proven [43] that the dropout layer help prevents the neural networks from overfitting. By it's nature, the idea of dropout is not complicated. It simply removes some of the nodes from the network during each training stage by setting them to zero. This forces the network to be redundant, meaning that it should provide the right classification even if some of the activations are dropped out, which makes sure that the network will not get overfitted to the training data. This layer is going to be used only during

the training process.

The last layer we are going to use is a softmax layer or a "loss" layer, which uses the softmax regression shown on the equation 2.10, to normalize the values. We are going to apply the softmax function to an input tensor, which will normalize the inputs in a way that the total sum of inputs is equal to 1. The shape of output of a softmax layer is the same as input - it simply normalizes the values. The output of the softmax layer can be interpreted as probabilities of our one-hot encoded labels.

$$S(z)_j = \frac{e^{z_j}}{\sum_{k=1}^{K} e^{z_k}} \quad \text{for} \quad j = 1, \ldots, K. \tag{2.10}$$

Now we need to choose the methods to evaluate the network performance. We are going to use cross-entropy together with ADAM [44] optimizer to minimize it. ADAM optimizer is a gradient-based optimization algorithm, which is based on the adaptive estimates. It is suited for the problems with large amount of data and a lot of parameters. The cross-entropy function (shown on the equation 2.11) is also known as a cost function - it measures how good is the classification. The higher the cost, the higher is the level of inaccuracy. The accuracy is calculated by the comparing the true values from the training labels with the results of the predictions from our network. The ultimate goal is to minimize the loss.

$$H(p, q) = -\sum_{x} p(x) \log q(x) \tag{2.11}$$

Model training

In the ideal case, we want to use all of the available data for every step of the model training, but since our dataset is so big, it's very expensive. Instead, we are going to use the batches randomly taken from the dataset. This approach is called stochaistic training, and it is very often used because it's very cheap, fast and provides good results.

We are going to perform the model training in iterations. For the final model the total amount of iterations used was 20000, which took over 8 hours of training

on my personal computer. The figure 2.7 shows the accuracy of the model with only 200 iterations where we could clearly see how even 200 iterations provide decent results.

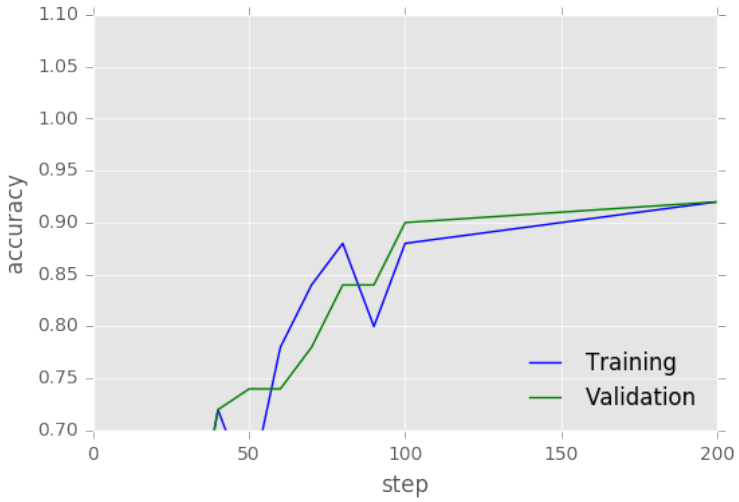

Figure 2.7: Accuracy after 200 training steps

During the each iteration, we will randomly pick the batch of data points from the training set and put it into the TensorFlow computational graph by replacing the described placeholders, which will be executed outside of the Python environment that is being used. During the iterations, occasionally we are going to check the accuracy on the next batch, to see how the overall accuracy of the model is being improved with each iteration. After the training, it's very important to store the trained model locally. It will make it possible to simply load the trained model into memory for further usage instead of training it every time, which is very costly in terms of time and computing.

Data preparation

Before attempting to classify the strokes received from the iPad, the raw data should be prepared and transformed for the further use of classification model. The example of raw stroke is shown on the figure 2.8. After parsing raw JSON into python classes we perform multiple calculations that we need to use for further analysis and feature extraction phase.

It's clear that the raw data we receive from the iPad is not compatible with the

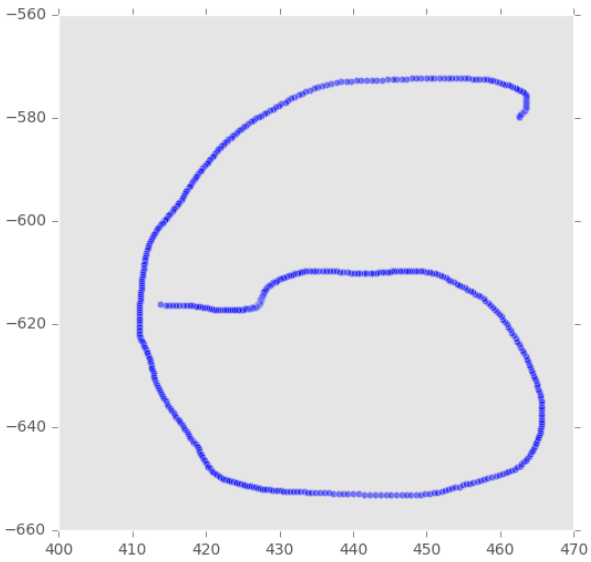

Figure 2.8: Raw stroke from the iPad

MNIST, as can be seen on the figure - the original stroke is much bigger and much thinner that any of the MNIST examples, because of the digital nature of the stroke.

If we compare the raw stroke with the MNIST sample shown on the figure 2.9, we can clearly see the difference. Here, the MNIST digit is shown as a two-dimensional matrix with the numbers between 0 and 1. This is the format of the digit that is used to train the deep convolutional neural networks. Which means that the raw strokes from the iPad should be transformed into the same format, to make it work with our model.

First, each stroke should be transformed into the 28x28 matrix, where each filled pixel corresponds to “1” and empty pixel to “0”. We know that the digit itself should fit 20x20 matrix, so first, we calculate the full width and height of the stroke, then select the maximum of them and calculate the ratio by dividing 20 to this maximum. After getting the ratio, we simply iterate through all the points in the stroke and multiply x and y axis values by the computed ratio. Then the coordinates of a digit are transformed into appropriate 28x28 matrix. The result of this transformation is shown on the figure 2.10, where it is seen that the stroke has been transformed into the matrix with an appropriate size and boundaries.

Next step is to make a stroke thicker. This will improve the accuracy of classifier, because in the MNIST dataset the digits are thick, taking several “pixels” in a

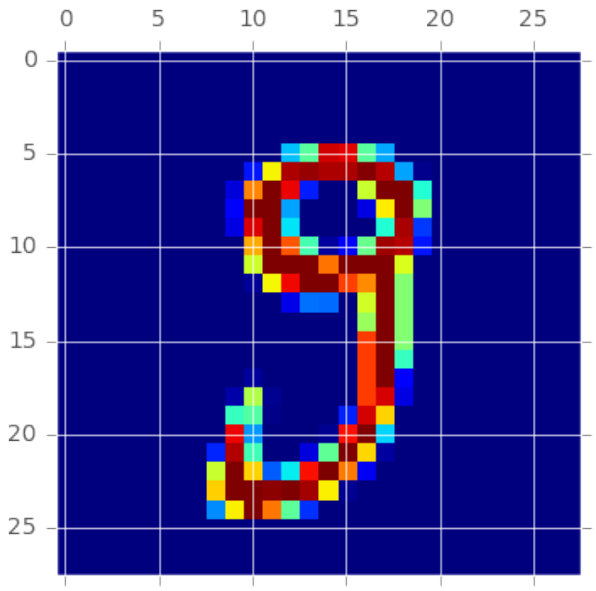

Figure 2.9: MNIST sample as a matrix

matrix. So we iterate over the matrix and for each "1" or the "pixel", we create four more pixels around it. We can allow such transformation, because it improves the classification rate of a digit and it does not modify the original stroke data which will be used for feature extraction. So it will now affect the feature extraction, but act as signal amplifier to improve the classification process.

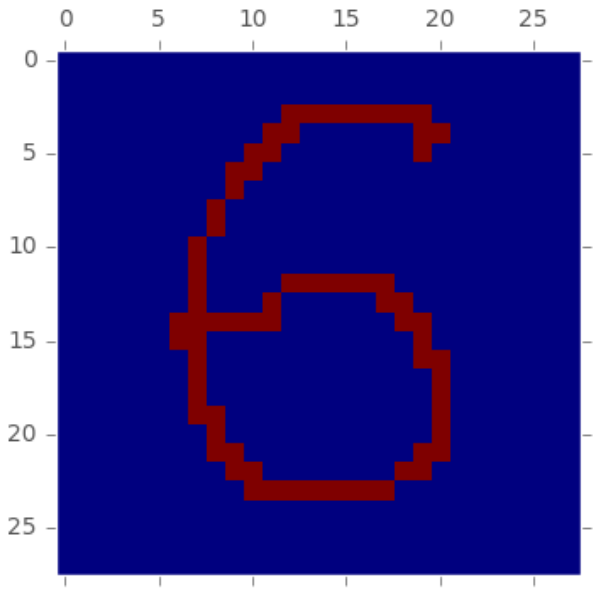

Figure 2.10: Matrix of transformed stroke

It is also possible to add noise around the stroke to mimic the MNIST digits even

more. Due to nature of MNIST digits and the fact that most of them are scanned images - the borders of digits are transparent to some degree. Since the data received from the iPad has digital nature - this noise is not present. In attempts to improve the classification rate the "noise generation" mechanism was also implemented. This mechanism would fill the empty pixels around the thin image with the random floating point values ranging from 0 to 1 to mimic the noise. But during the latest stages of development and measuring the accuracy of the models, it was clear that using the thickening instead of noise provided much better classification results.

The last step before using classification is to center the strokes by the center of mass of their pixels. It is necessary because all of the MNIST digits in the dataset are positioned according to their center of the mass. This step is quite important because skipping it might lead to the situations when the classifier, due to it's weigths-based approach will pay more attention to the position of the stroke rather than it's shape. One example could be the positioning of digit 1. If we place digit 1 in the boundary box without aligning it by the center of it mass - it will be aligned closer to the right boundary and the classification may result in interpreting digit 1 as digit 4, because the vertical stroke in digit 4 is in the same way aligned closer to the right boundary.

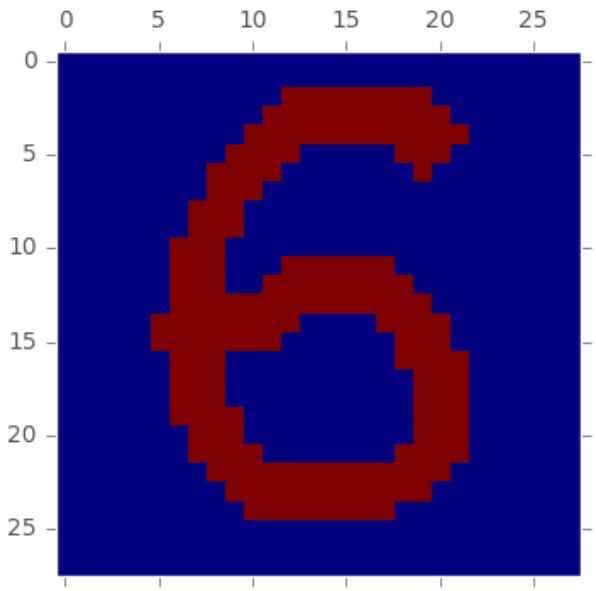

Figure 2.11: Thick matrix

After each stroke of the drawing is transformed into the appropriate format, next

step is to load previously saved TensorFlow model into memory and try to classify each stroke of the drawing.

Most probable elements

There are several digits that usually are written with multiple strokes by the most of the people. Most popular multiple-stroke digits are 4, 5, 7. But in reality, any digit can be written with multiple strokes and this is a big challenge.

To identify such strokes, the combination of several mechanisms are being used. First mechanism is focused on finding the pairs of strokes with the shortest pause between them. In order to do that, we calculate the total duration of "drawing time" and the average duration of a single stroke. Then we calculate the total "pausing time" and the average pause time between the strokes. Next step is to calculate the pauses between all of the combinations of strokes and select the ones that are two times faster than average pause duration, those are our candidates for possible elements.

Second mechanism is focused on finding the pairs of strokes with the shortest distance between them. Since each stroke might consist from the hundreds of points, then finding the closest points between the strokes and measuring distance between them might be very costly in terms of resources and time. The simplified approach that is implemented is identifying the central or "weight" point of each stroke. In this way, the stroke that consists of hundreds of points is represented by the single point. Next step is to calculate the average distance between the strokes and find the combinations of strokes, whose distance is four timed shorter then the average, those are also our candidates for possible elements.

Third mechanism is focused on using k-means clustering to find the strokes in the same clusters. Here we face the similar issue as in previous approach, where in case of each stroke consist of hundreds or even thousands of points, then using all of them in k-means will be very costly. In order to optimize the process - i simplify each stroke by using evenly distributed 10% of their points. This way the whole stroke is properly represented by the only one tenth of the points, which requires ten times less calculations. The k-means is ignoring the circle and clock hands, it

is using the k-value 12, which should capture 12 numbers around the clock. The result of the k-means clustering is shown on the figure 2.12, where it is seen how the strokes are getting clustered.

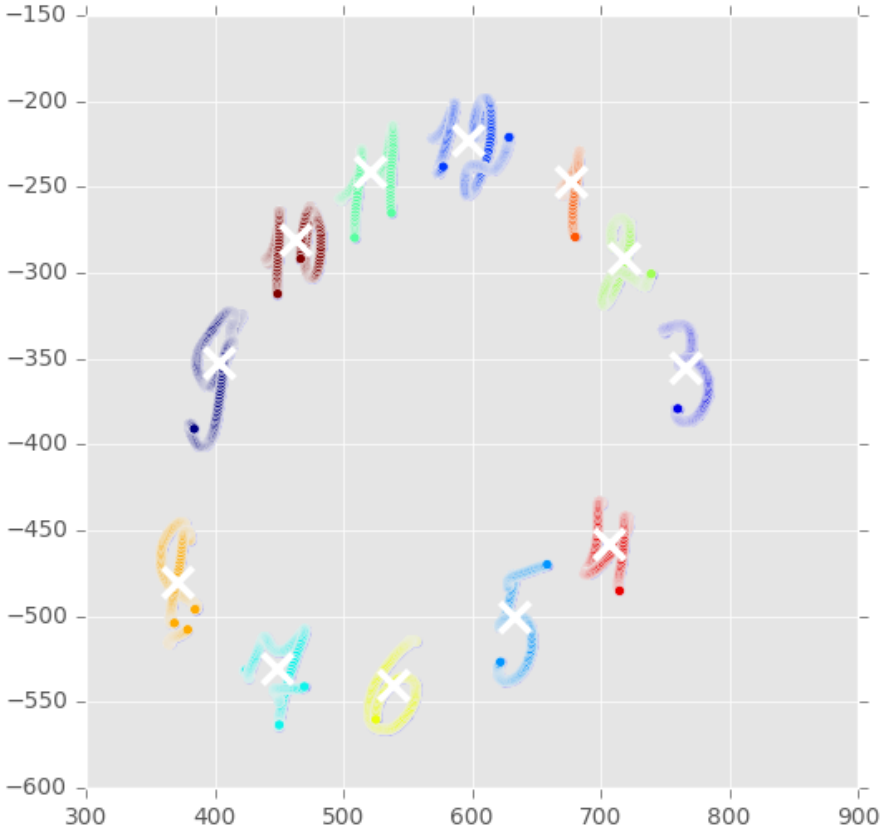

Figure 2.12: K-means clustering

If the combination of strokes is identified by at least 2 out of 3 mechanisms - those strokes are considered to be multiple-stroke digits. But in most of the situations, when there are no additional elements on the drawing - it is enough to use only k-means clustering to successfully identify multi-stroke digits.

2.5.5 Digit classification

After the possible digits are clustered and grouped, it's time to classify the digits using the combination of heuristics and our deep convolutional neural network trained on a MNIST dataset.

Since one of the main goals of this project is to be able to identify the mild-cognitive impairments, in the initial implementation it is assumed that the clocks

will be drawn more or less correctly - in a sense that there will be no repeated digits and the digits will be drawn in the appropriate places. Because the problem of identifying the repeating or incorrect digits is quite big & complicated and deserves the separate master thesis on its own.

So the classification algorithm assumes that existing 12 clusters represent the twelve digits on the clock-face. And because of the reasons above it also expects that out of the 12 clusters, there will be no repeating or crossed-out digits. Which means that as long as those assumptions hold, the results of classifications will be accurate, but in case if assumption is violated - the result is not guaranteed to be accurate.

Classification algorithm starts with going over the clusters which have only single stroke and trying to classify each stroke using the neural network. The results of classification are stored with reference to the cluster. The results are one-hot encoded, which means that single result contains the probabilities of a stroke being each digits from 0 to 9, where the sum of probabilities is always equals to one.

After classifying the single-stroke clusters, the algorithm goes thorough the multi-stroke clusters and for each cluster it merges the strokes inside of a cluster so they would be used as a single input for the neural network and tries to classify this stroke as a single digit. The results of classification are also stored with a reference to the cluster.

Next step is to iterate over multi-stroke clusters again, but this time to classify each stroke inside of a cluster individually and store those results in a separate collection for further comparison. This step is essential for classifying the digits such as 10, 11 and 12. Because our neural network knows nothing about those digits. (Since it was trained only on the digits from 0 to 9)

Then the classification algorithm goes through all of the classification results and tries to compare them and determine the the final result. It finds the first occurrence for each of the digits from 1 to 12 and assigns it to be the final results. If afterwards there is a second candidate for the same digit, the algorithm will compare their probabilities of it being the same digit and will choose the one with higher

probability. The candidate with lower probability is treated as "misclassified" and analyzed further - the second highest probability is taken from the one-hot encoded results and evaluated, whether this digit is already classified or not. It both first and second highest probabilities are already classified with the higher probability - the stroke is considered to be misclassified and stored separately.

By the time the algorithm reaches this stage, all of the digits from 1 to 9 should be classified. The only misclassified clusters that are left - are the ones that had the low probabilities of being a digits from 0 to 9, which are the digits 10, 11 and 12 respectively. During this stage the algorithm tries to compare the results of classifying the cluster as a single stroke with the results of classifying the same cluster stroke by stroke. If the stroke by stroke classification yields 1 and 0 with mean probability higher than the single stroke classification, the cluster is considered to be number 10. The similar comparisons are invoked for the digits 11 and 12.

2.5.6 Digit classification challenges

During the implementation phase i came to the realization that using only convolutional neural networks for digit classification is not enough to achieve desired results for classifying the strokes in a digital clock drawing test.

Since before starting the work on thesis i had no previous experience working with convolutional neural networks and lack of knowledge about how they work, i have made a false assumptions about the way they will perform. My false assumption was that if the stoke doesn't look like any digit - it will not be recognized as one.

During the planning phase i thought that the classification should work in a way that i could classify all of the strokes related to the digits using the neural networks and consider the rest of the strokes to be related to the other parts of the clock. But as it turns out - this approach is not possible, because of the way neural network is trained. I could not use the neural network to filter out the strokes that are not related to digits.

To show this problem, let's consider an example of single horizontal stroke shown on the figure 2.13. If we, as a humans were asked to classify this stroke as a digit

- we would probably say that this is not a digit, it's just a line which can be part of some digit. But since the neural network was trained on the digits from 0 to 9, those 10 digits are everything this network knows about, so it will try to assign a probability of a stroke being a certain digit according to weights to any type of stroke.

The horizontal stroke from the example on figure 2.13 is classified as a digit 7 with the probability of 0.99, because of the distribution of weights in the trained model, and this is expected behaviour that i was not aware of. And this fact is what makes the stroke classification complicated with using the same approach as the one chosen for this thesis.

This is the reason why using only convolutional neural network was not enough for accurate stroke classification. But the combination of using geometrical positions, heuristics and other machine learning techniques such as k-means clustering together with convolutional neural networks produce much better results.

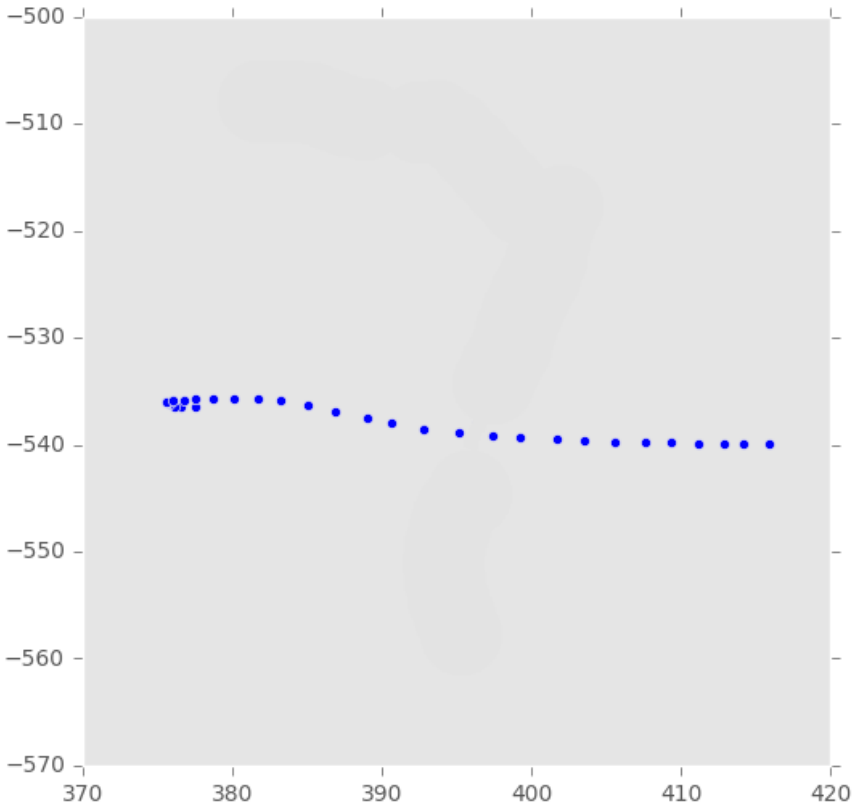

Figure 2.13: Example of the horizontal stroke

2.6 Deployment

The Internet is full of tutorials about how to train a neural network classifier, how to build random forest, deep convolutional network, and so on. But what is really hard to find is the information and tutorials about how to deploy the model into production.

There is a huge gap between training the model on the local machine that is capable of classifying the digits, and deploying this model together with web application to run in production environment. This is also a problem of performance and possibility to invoke the models.

For example the model created in MATLAB or Octave can work in the native environment, but there is no way to run this code on the web server. And even if it is possible - the code will not be optimized. The main problems is - how to create a model in a way that it will work language agnostic and work as a function?

Ideally, the model should work in a way that it will accept some input and produce the output. The closest solution to this is to wrap the model with the web-server or use the technology such as Amazon AWS Lambda to run the model inside a function.

In scope of this thesis several machine learning models should be deployed and used production environment. Some of the models were created using the open-source Python library called scikit-learn and convolutional neural network model using the open-source TensorFlow framework.

In case of scikit-learn models, the Python's built-in persistence model called "pickle" was used for storing and re-using the models. Pickle represents powerful algorithm for serializing and de-serializing Python object structure into files.

In case of TensorFlow, the tf.train.Saver object provided by the library was used for saving and restoring the model. The Saver object provides a way to save the specified list of variables in the TensorFlow computational graph. It also provides a means to restore the saved variables, and possibility to create and load checkpoints.

Since most of the code was written using the Python language, the simplest Python web framework with the name "Flask" was chosen for wrapping the machine learning models and using them from the web server. Flask application initializes the web server and during the incoming requests loads the models into memory and uses them to produce the results.

The Flask web application was deployed to the AWS EC2 instance using the AWS Elastic Beanstalk orchestration tool, which handles almost all of the infrastructure issues such as load-balancer configuration, virtual machine configuration and so on. Thus making the deployment process much easier and faster.

Chapter 3

Analysis

3.1 Feature extraction

Feature extraction is a very important part of digital clock drawing test implementation and analysis. General idea is to represent the whole drawing as set of values. The more features are extracted, the better - because it's impossible to know in advance which features are significant and which are not, we can only make assumptions.

But extracting many features and than performing statistical analysis might shed some light on the features that require more attention. Finding interesting and unusual features is quite challenging. Currently, in the initial version of clock drawing test, 373 features are extracted from the single drawing. Groups of features are described in the following sections.

3.1.1 Base features

After we've classified the strokes on the drawing. Next step is to extract the features from them. First phase of feature extraction is to calculate the base drawing features, which can be done even before classification. During the data parsing process, some of the base features are already being calculated, because since we already have a need to iterate over all of the points present in the drawing - it's good to fit some calculations into the processing to not waste resources later.

Some of the basic features are extracted on a per stroke basis during the initial

data parsing and they include things like the ones shown in the table 3.1.

Feature	Description
Stroke width	Width of the stroke in points (x - axis)
Stroke height	Height of the stroke in points (y-axis)
Stroke area	Stroke width multiplied by stroke height
Stroke length	Number of points in the stroke
Stroke duration	Total duration in milliseconds of time it took to draw the stroke

Table 3.1: Example base features

Basically, during this phase the features related to the stroke boundary are calculated. They are going to be heavily used later, during the more advanced feature extraction.

In general, features are coming from different layers, and they all should be combined together to represent the whole drawing - so the main idea is to represent the drawing as a set of features. First layer of features consists of each point's feature, since each point has attributes such as pressure, time, x and y axis - those are the point's features.

Next layer is a stroke layer. Each stroke is a collection of many points, so many of the stroke features are calculated using averages of the stroke's points. For example the mean stroke pressure is calculated based on the pressure of it's points. Together with the mean, the variance, standard deviation and other characteristics are also calculated.

The third layer is a drawing layer. Since drawing is a collection of strokes - most of it's the features are calculated based on the features of strokes the same way as stroke features are calculated based on points. Few examples of drawing features are described in table 3.2.

Feature	Description
Total time	Amount of milliseconds it took to perform a test
Total pause time	Amount of milliseconds spent on thinking
Total draw time	Amount of milliseconds spent on drawing
Thinking/doing ratio	Ration of time spent thinking versus drawing
Average pressure	Mean value of pressures across the strokes
Pressure standard deviation	Value of pressure standard deviation across the strokes

Table 3.2: Example drawing features

Other features include things such as longest and shortest pauses, longest and shortest strokes, averages and distributions of altitude angles of the pen as well as azimuth angles, and so on.

3.1.2 Circle features

After identifying the base drawing features, we move on to using the classification results to extract more features from the drawing. After classification, we know to a certain degree, where do we have each part of the clock - so that should be used for further calculations.

One of the first elements of the clock that gets classified is a contour around the clock face, which is called the circle. Some of the circle features are shown in the table 3.3. Few important circle features were also already described in the section about circle classification.

3.1.3 Clock hands features

There are several versions of clock drawing test, some of them pay special attention to the clock hands, because the patient is asked to draw the clock showing certain

Feature	Description
Direction	Direction of the stroke (CW or CCW)
Beginning quadrant	Quadrant where the circle was started
Ending quadrant	Quadrant where the circle was ended
Crossed quadrants	Number of quadrants crossed during the drawing
Radius of "perfect" circles	Inner, outer and center circles radius
Circumference of "perfect" circles	Inner, outer and center circles circumference
Mean square errors	Errors against "perfect" circles

Table 3.3: Example circle features

time of the day, which is usually 10:10. In this case it is important to check that the patient has drawn correct clock hands.

But since in our initial version of digital clock drawing test the patient is not asked to draw a certain time, clock hands are not heavily analyzed - only the basic features together with motion mass parameters are extracted from the clock hands together with few additional features like:

- Number of strokes in clock hands
- The lengths of clock hands
- The ratio between hour and minute hand

3.1.4 Digit features

Extracting digit features is quite straightforward. To keep the number of features consistent it is always assumed that the clocks should have twelve digits, from 1 to 12. And the digit features are calculated for each number. After each of those digits is classified, the same features as were extracted from each stroke are separately extracted for each digit. Here are few example:

- Total time it took to draw the digit 6 (and all other digits)
- The width-height ratio of digit 3 (and all other digits)
- The standard deviation in pressure of digit 7 (and all other digits)
- Width of digit 5 (and all other digits)
- Area of digit 11 (and all other digits)
- Number of points in digit 1 (and all other digits)

In cases when digit consists of several strokes, in the initial implementation of clock drawing test those strokes are merged together in a single stroke, and during feature extraction they are treated with a single stroke. But the number of strokes in a digit is also a separete feature.

3.1.5 Motion mass features

Motion mass parameters can be measured on a per-stroke basis. First, the motion mass features are extracted from the strokes of the whole drawing, so that those parameters could characterize the drawing as a whole.

Next, the motion mass parameters are calculated for each part of the clock. Which means that if some element of the clock is drawn using several strokes - those strokes are merged together and are considered to be a single stroke.

Motion mass parameters include the following:

- Length of trajectory of the stroke
- Velocity mass of the stroke
- Acceleration mass of the stroke
- Angle mass of the stroke
- Jerk mass of the stroke
- Average velocity of the stroke

- Average acceleration of the stroke
- Average slope of the stroke
- Average jerk of the stroke

3.1.6 Output format

After all of the features are extracted, they are stored in a Python dictionary. By default, the dictionary is transformed into JSON format, where the short name of the feature is used as a key and feature value used as a value.

Each type of features have a corresponding prefixes (e.g. "circle_" or "digit_6_") to make them identifiable. The CSV format is also supported. In case of CSV output, all of the features are returned as coma-separated values. The results of feature extraction could be used for analysis or for machine learning models training.

3.2 Statistical analysis

After gathering the data and extracting the features from the data we will have two separate datasets of features. One dataset consists of features of the drawings of the control individuals and another dataset consists of features of the drawing of the cognitively impaired patients.

First step in the analysis is a comparison of means of two datasets. We take the collection of values of a single feature from both datasets and compare them to identify whether they belong to the same population. The hypothesis tests [45] should be used to test the validity of a claim about a population.

The null hypothesis in our case is that there is absolutely no difference between two datasets. The alternative hypothesis in our case is that there is a significant difference between the populations of two datasets.

There are several major types of means comparison tests, which include one-sample test, two independent sample test and repeated measure test. In the thesis work, we are going to use two independent sample test, because that's the one most suitable for our case. In this test, we use two independent samples of data to test whether there is any difference in means between populations

The p-value is being used as a measure of significance of comparison results. Or, more specifically, two-sided p-value, which implies that the hypothesis test is performed without any directionality.

One of the most popular ways to measure p-value is to use t-statistic tests. The most commont t-test is a Student's t-test [46], which is not suitable in our case, because the sample sizes might be not equal. That's why the Welch's t-test [47] was chosen - it allows unequal sample sizes.

Apart from finding the most significant features, the features extracted during the clock test should also be analyzed for correlation. If some features have really high correlation coefficient - it means that they should not be used together during the machine learning model training, because they could impair the results of algorithm's learning. One of the most popular ways to measure the correlation coefficient is to

use the Pearson correlation coefficient [48] calculation.

When enough data is received from the pilot study in the hospital, all of the extracted features should be thoroughly analyzed before creating machine learning models. The mean comparison should help select the most significant features from the whole set of features. And with correlation coefficient calculation the highly correlated features should be filtered out in order to not be used together. This could be achieved by having many trials of model training with different features, to see which combinations of features produce the best results.

For calculating the p-value and the Pearson correlation coefficient the open-source Python library "SciPy" was used.

3.3 Example of mean comparison

The organization of pilot study with real patients is still an ongoing process, so there is yet no data with the results of tests, performed by cognitively impaired patients. And without this data it is obviously impossible to compare the means between healthy and cognitively impaired patients, and to analyze them.

Although, during the testing phase of the implementation, the data from several control subjects was acquired. For the sake of example, the control group's data was divided into two separate datasets with equal size of 7 subjects, based on their age. One dataset has the samples from subjects under the age of 35, and another those over this age. The results of the mean comparison are shown on the table 3.4.

Feature	p-value	t-statistic
digits_11_height_width_ratio	0.0972	-2.9693
digits_9_mm_jerk_mass	0.0976	-2.3802
digits_9_mm_acc_mass	0.1057	-2.2931
digits_9_mm_velocity_mass	0.1088	-2.2608
digits_4_mm_velocity_mass	0.1619	-1.7129
digits_2_height	0.2006	-1.4734
digits_12_mm_avg_jerk	0.2232	1.7446
circle_center_diameter	0.2731	-1.2308
digits_9_mm_avg_acc	0.2972	-1.2587
digits_9_mm_angle_mass	0.3136	1.2078
digits_6_height	0.3193	-1.1361

Table 3.4: Mean comparisong of two datasets

If we set our threshold of significance as 10%, then, as can be seen from the table, there are only two significant features. One of them is "digits_11_height_width_ratio" which stands for the ratio between height and width of the digit eleven. The second feature is "digits_9_mm_jerk_mass" which stands for the jerk mass or the combined rate of change of acceleration of the digit 9.

Since the sample size is so tiny, then those results should not be considered seriously, it is just an example of how implemented code can compare two datasets and find the features that can distinguish those datasets. Plus division of the data into two datasets was based on the age solely for reasons to have both datasets of equal size, not taking anything else into account.

Even though it is just an example, it clearly shows that using the described digital clock drawing test implementation, the very subtle differences can be used for diagnosis. It is impossible for clinician's eye to precisely measure the combined rate of acceleration of the digit 9, or to notice this difference between patients. But the machine can notice it, and use it in decision making.

The more real data is acquired, the more precise the results of mean comparison will be. And those results should be used for feature selection during the training of machine learning algorithms.

3.4 Further analysis

Initially, it was planned to conduct the pilot study in scope of this thesis work, but the organization of the study took longer than expected. The plan was to implement the digital clock drawing test, then perform pilot study, acquire the data and perform initial statistical analysis on the data and train several machine learning algorithms as a proof of concept. All of the tools and the code for analysis was implemented, but since the real data was not yet acquired, the analysis was not done yet.

At the moment of this writing the digital clock drawing test implementation is ready to be used in the real-life medical facility environment. Although it has limited functionality, but it is already useful. It has also been accepted by the clinicians during our meetings and we are ready to gather the data and perform statistical analysis.

In the process of development, numerous meetings with the interested clinicians were organized and their feedback and requirements were discussed and implemented. The feedback was mostly about the user interface of the iPad application, but also about the security, data integrity and about using the cloud solution for storing the data instead of offline solution.

Currently we are in the process of scheduling the pilot study in the hospital. The organizational part of the study with real patients took longer and was more complicated than we have anticipated in the beginning of the thesis work. But the work continues, and soon the data for the analysis should be ready.

After conducting the pilot study and acquiring the real data, the tools developed for statistical analysis will be used and the data will be thoroughly analyzed. After gathering significant amount of data, next step would be to start using machine learning techniques to create models based on the acquired data.

Chapter 4

Discussion

Since the process of scheduling the pilot study in the hospital is ongoing due to fact that the organizational part of the study with real patients took longer than was anticipated in the beginning, the amount of data gathered from the control individuals in scope if current work is unarguably inefficient for making definitive conclusions and using the current version of developed digital clock drawing test in the hospitals for cognitive impairment detection and diagnosis at this stage. But it can already be used for providing valuable feedback for the clinicnans and for data acquisition.

Therefore this work should be treated as foundation, that further research should be based on, and considered as a pilot study with the aim of understanding whether it is worth moving forward in this area. And this work clearly indicates that such developments can be useful in medical facilities, even with limited functionality. So the further research and implementation of digital clock drawing test should be continued.

As it was mentioned in the section about classification, the initial clock drawing test implementation developed in scope of this thesis focuses mostly on the detection of mild cognitive impairments, so to some degree the algorithms expect the drawing to be more or less correct. Meaning that if there are any repeated digits or if the positioning of digits is entirely incorrect - the classification will most probably fail.

Another limitation of presented implementation is that it focuses on the recognition of main elements of the clock, which include clock circle, clock hands and clock

digits. So if any other elements are present, such as marks or crossed-out digits, or the drawing is greatly distorted - the classification will also fail in this case.

Yet another drawback of current implementation is it's optimization. There are many parts of the code that are not dependant on each other - so it leaves the room for improvement in a sense of parallelization of the code. It should be possible to separate the code workflow in separate independent executions and run them in parallel to improve overall speed of execution.

As it was also discussed in the digit classification problems section, there are several drawbacks of using the deep convolutional neural network for stroke recognition. It is a great tool for digit recognition, but in case of stroke recognition - using it alone is not enough, due to the way it works and the fact that it was trained to only recognize the digits from 0 to 9.

It should be possible to greatly improve stroke classification results by training the models not only on digits from the MNIST dataset, but also on the digits 10, 11 and 12. In the ideal scenario - after having many examples of clock drawings, the classification model should be trained to directly classify the parts of the clock. But it requires a lot of data from real patients.

One of the main challenges in using the machine learning techniques for medical diagnostics is the fact that the results must be interpretable by the clinicians. Most of the machine learning techniques work as a black-box in a sense that they might correctly distinguish cognitively impaired from healthy individual, but they will fail to give any kind of reasonable explanation about the diagnosis.

There is a clear trade-off between the accuracy and the interpretability of the results. And at this stage when this software works as an aid for the clinicians - we should sacrifice the accuracy for interpretable models. For this reasons the machine learning algorithms such as Decision trees, Random Forests and especially Bayesian Rule Lists and Supersparse Linear Integer Models should be preferable for diagnosis, because the reasons behind their diagnosis can be understood by the clinician.

Even though the version of digital clock drawing test in scope of presented thesis is not completely finished and has a lot of open issues, it is still can be used in medical

facilities and provide limited, but useful results even today. The solid foundation for making digital versions of popular manual tests has been also developed and tested in scope of this work. The iPad application is stable and refined. The back-end infrastructure is in place and running smoothly. The statistical analysis tools are ready to be used once the data is acquired.

Chapter 5

Conclusion

The main goal of this thesis was initial implementation of digital clock drawing test that could be used and would provide reasonable results outside of the academic environment. Side goal was to analyze the pilot study results to determine whether it is possible to differentiate healthy and cognitively impaired patients based on the extracted features and is it worth moving forward with further digital clock test implementation.

In the beginning, the whole thesis work was based on the existing initial implementation of common infrastructure for performing different digital versions of common manual pen and paper tests, which was already in place. This infrastructure included the very basic iPad application and few simple back-end services for receiving and storing the data. But this infrastructure was not sufficient for the actual needs.

Therefore, in the process of thesis work, apart from implementing the digital clock drawing test itself, the whole infrastructure was overhauled and vastly improved to be more robust and reliable. This includes iPad application development, back-end services development and change in the architecture and technology of the infrastructure. The development based on the feedback and requirements from the clinicians was also implemented and tested.

The clock drawing test implementation was created from scratch. The whole process of receiving the data from the iPad, parsing and processing the raw data into appropriate format, classifying the strokes and extracting features from identified

clock elements was done in scope of the thesis. In the beginning, most of the code was written in Jupyter notebooks for faster prototyping, but during the latest stages the whole code was significantly refactored and moved into separate Python modules to work on a webserver in production environment.

Major part of the thesis includes digit classification which was achieved without using any existing character recognition systems. The whole digit classification mechanism was developed from scratch in scope of the thesis using the TensorFlow framework. The deep convolutional neural network for digit recognition was created for recognizing the numbers on the clock. Training, evaluating and improving the TensorFlow model for digit recognition is a significant part of the project. The problems and limitations of this approach were also investigated and explained in the thesis.

After digital clock drawing test implementation was developed, the mechanism for statistical analysis was also implemented. This includes the necessary code for the mean comparison and the correlation analysis of the data. The questions and problems encountered during the present thesis work were posted and explained in the discussion section.

Overall, results achieved in this thesis together with implemented infrastructure may not only play an important role in the further research and studies, but also could already be used in the real-world medical facilities. The achieved results clearly indicate that it it worth moving forward with digital clock drawing test implementation. Although, the version of digital clock drawing test implemented in scope of the thesis is far away from being final version, and it provides only limited results and in limited settings, but it is still already useful. There is a big room for improvement in stroke classification, machine learning methods, overall performance and so on. But this require gathering much more data and constantly iterating and improving the code.

Acknowledgements

The present work would be impossible without the cooperation with Dr. Toomas Toomsoo from the East-Tallinn Central Hospital. Who have organized the meetings with the patients for the data acquisition.

This thesis was also partially supported by Research Funding Project of the Tallinn University of Technology - B37.

Bibliography

[1] Kirk A and Kertesz A. On drawing impairment in alzheimerś disease. *Archives of Neurology*, 48(1):73–77, 1991. doi: 10.1001/archneur.1991.00530130083024. URL + http://dx.doi.org/10.1001/archneur.1991.00530130083024.

[2] Jonas Jardim de Paula, Debora Marques de Miranda, Edgar Nunes de Moraes, and Leandro Fernandes Malloy-Diniz. Mapping the clockworks: what does the clock drawing test assess in normal and pathological aging? *Arquivos de Neuro-Psiquiatria*, 71:763 – 768, 10 2013. ISSN 0004-282X.

[3] Yurinosuke Kitabayashi, Hideki Ueda, Jin Narumoto, Kaeko Nakamura, Hitoshi Kita, and Kenji Fukui. Qualitative analyses of clock drawings in alzheimer's disease and vascular dementia. *Psychiatry and Clinical Neurosciences*, 55(5): 485–491, 2001. ISSN 1440-1819. doi: 10.1046/j.1440-1819.2001.00894.x. URL http://dx.doi.org/10.1046/j.1440-1819.2001.00894.x.

[4] Lynnette Pei Lin Tan, Nathan Herrmann, Brian J. Mainland, and Kenneth Shulman. Can clock drawing differentiate alzheimer's disease from other dementias? *International Psychogeriatrics*, 27(10):1649–1660, 10 2015. doi: 10.1017/S1041610215000939.

[5] Luigia Trojano and Guidoc Gainotti. Drawing disorders in alzheimer's disease and other forms of dementia. *Journal of Alzheimer's Disease*, 53(1):31–52, 2016. doi: 10.3233/JAD-160009. URL https://www.ncbi.nlm.nih.gov/pubmed/27104898.

[6] H. Tuokko, T. Hadjistavropoulos, J. A. Miller, and B. L. Beattie. The clock test: A sensitive measure to differentiate normal elderly from those with alzheimer disease. *Journal of the American Geriatrics Society*, 40(6):579–

584, 1992. ISSN 1532-5415. doi: 10.1111/j.1532-5415.1992.tb02106.x. URL http://dx.doi.org/10.1111/j.1532-5415.1992.tb02106.x.

[7] C. C. Price, H. Cunningham, N. Coronado, A. Freedland, S. Cosentino, D. L. Penney, A. Penisi, D. Bowers, M. S. Okun, and D. J. Libon. Clock drawing in the montreal cognitive assessment: Recommendations for dementia assessment. *Dementia and Geriatric Cognitive Disorders*, 31(3):179–187, 2011. ISSN 1420-8008. URL http://www.karger.com/DOI/10.1159/000324639.

[8] Barbara Costa Beber, Renata Kochhann, Bruna Matias, and Marcia Lorena Fagundes Chaves. The clock drawing test: Performance differences between the free-drawn and incomplete-copy versions in patients with mci and dementia. *Dementia & Neuropsychologia*, 10:227 – 231, 09 2016. ISSN 1980-5764.

[9] Marilyn S. Albert, Steven T. DeKosky, Dennis Dickson, Bruno Dubois, Howard H. Feldman, Nick C. Fox, Anthony Gamst, David M. Holtzman, William J. Jagust, Ronald C. Petersen, Peter J. Snyder, Maria C. Carrillo, Bill Thies, and Creighton H. Phelps. The diagnosis of mild cognitive impairment due to alzheimer's disease: Recommendations from the national institute on aging-alzheimer's association workgroups on diagnostic guidelines for alzheimer's disease. *Alzheimer's & Dementia: The Journal of the Alzheimer's Association*, 7(3):270–279, 2017/03/03 XXXX. ISSN 1552-5260. doi: 10.1016/j.jalz.2011.03.008. URL http://dx.doi.org/10.1016/j.jalz.2011.03.008.

[10] Roberto Alves Louren, Sergio Telles Ribeiro-Filho, Irene de Freitas Henriques Moreira, Emylucy Martins Paiva Paradela, and Aline Sobral de Miranda. The clock drawing test: performance among elderly with low educational level. *Revista Brasileira de Psiquiatria*, 30:309 – 315, 12 2008. ISSN 1516-4446.

[11] Justin A. Nyborn, Jayandra J. Himali, Alexa S. Beiser, Sherral A. Devine, Yangchun Du, Edith Kaplan, Maureen K. O'Connor, William E. Rinn, Helen S. Denison, Sudha Seshadri, Philip A. Wolf, and Rhoda Au. The framingham heart study clock drawing performance: Normative data from the offspring cohort. *Experimental Aging Re-*

search, 39(1):80–108, 2013. doi: 10.1080/0361073X.2013.741996. URL http://dx.doi.org/10.1080/0361073X.2013.741996. PMID: 23316738.

[12] Mary C. Lessig, James M. Scanlan, Hamid Nazemi, and Soo Borson. Time that tells: critical clock-drawing errors for dementia screening. *Int Psychogeriatr*, 20(3):459–470, Jun 2008. ISSN 1041-6102. doi: 10.1017/S1041610207006035. URL http://www.ncbi.nlm.nih.gov/pmc/articles/PMC2704110/. 17908348[pmid].

[13] Soo Borson, James Scanlan, Michael Brush, Peter Vitaliano, and Ahmed Dokmak. The mini-cog: a cognitive 'vital signs' measure for dementia screening in multi-lingual elderly. *International Journal of Geriatric Psychiatry*, 15(11):1021–1027, 2000. ISSN 1099-1166. doi: 10.1002/1099-1166(200011)15:11¡1021::AID-GPS234¿3.0.CO;2-6.

[14] Donald R Royall, Jeffrey A Cordes, and Marsha Polk. Clox: an executive clock drawing task. *Journal of Neurology, Neurosurgery & Psychiatry*, 64(5):588–594, 1998. ISSN 0022-3050. doi: 10.1136/jnnp.64.5.588. URL http://jnnp.bmj.com/content/64/5/588.

[15] Mario F. Mendez, Thomas Ala, and Kara L. Underwood. Development of scoring criteria for the clock drawing task in alzheimer's disease. *Journal of the American Geriatrics Society*, 40(11):1095–1099, 1992. ISSN 1532-5415. doi: 10.1111/j.1532-5415.1992.tb01796.x. URL http://dx.doi.org/10.1111/j.1532-5415.1992.tb01796.x.

[16] Kenneth I. Shulman, Dolores Pushkar Gold, Carole A. Cohen, and Carla A. Zucchero. Clock-drawing and dementia in the community: A longitudinal study. *International Journal of Geriatric Psychiatry*, 8(6): 487–496, 1993. ISSN 1099-1166. doi: 10.1002/gps.930080606. URL http://dx.doi.org/10.1002/gps.930080606.

[17] Peter J. Manos and Rae Wu. The ten point clock test: A quick screen and grading method for cognitive impairment in medical and surgical patients. *The International Journal of Psychiatry in Medicine*,

24(3):229–244, 1994. doi: 10.2190/5A0F-936P-VG8N-0F5R. URL http://dx.doi.org/10.2190/5A0F-936P-VG8N-0F5R. PMID: 7890481.

[18] Trey Sunderland, James L. Hill, Alan M. Mellow, Brian A. Lawlor, Joshua Gundersheimer, Paul A. Newhouse, and Jordan H. Grafman. Clock drawing in alzheimer's disease. *Journal of the American Geriatrics Society*, 37(8):725–729, 1989. ISSN 1532-5415. doi: 10.1111/j.1532-5415.1989.tb02233.x. URL http://dx.doi.org/10.1111/j.1532-5415.1989.tb02233.x.

[19] Joella E. Storey, Jeffrey T. J. Rowland, David Basic, and David A. Conforti. Accuracy of the clock drawing test for detecting dementia in a multicultural sample of elderly australian patients. *International Psychogeriatrics*, 14(3): 259–271, 009 2002. doi: 10.1017/S1041610202008463.

[20] James M. Scanlan, Michael Brush, Christina Quijano, and Soo Borson. Comparing clock tests for dementia screening: naïve judgments vs formal systems—what is optimal? *International Journal of Geriatric Psychiatry*, 17(1):14–21, 2002. ISSN 1099-1166. doi: 10.1002/gps.516. URL http://dx.doi.org/10.1002/gps.516.

[21] Joella E. Storey, Jeffrey T. J. Rowland, David Basic, and David A. Conforti. A comparison of five clock scoring methods using roc (receiver operating characteristic) curve analysis. *International Journal of Geriatric Psychiatry*, 16(4):394–399, 2001. ISSN 1099-1166. doi: 10.1002/gps.352. URL http://dx.doi.org/10.1002/gps.352.

[22] W. Souillard-Mandar, R. Davis, C. Rudin, R. Au, and D. Penney. Interpretable Machine Learning Models for the Digital Clock Drawing Test. *ArXiv e-prints*, June 2016.

[23] M. Wang, B. Wang, J. Zou, J. Zhang, and M. Nakamura. Quantitative evaluation of hand movement in spiral drawing for patients with parkinson's disease based on modeling in polar coordinate system with varied origin. In *The 2011 IEEE/ICME International Conference on Complex Medical Engineering*, pages 169–173, May 2011. doi: 10.1109/ICCME.2011.5876726.

[24] S. Nõmm, A. Toomela, J. Kozhenkina, and T. Toomsoo. Quantitative analysis in the digital luria's alternating series tests. In *2016 14th International Conference on Control, Automation, Robotics and Vision (ICARCV)*, pages 1–6, Nov 2016. doi: 10.1109/ICARCV.2016.7838746.

[25] S. Nõmm, K. Bardõš, I. Mašarov, J. Kozhenkina, A. Toomela, and T. Toomsoo. Recognition and analysis of the contours drawn during the poppelreuter's test. In *2016 15th IEEE International Conference on Machine Learning and Applications (ICMLA)*, pages 170–175, Dec 2016. doi: 10.1109/ICMLA.2016.0036.

[26] Jamie Cohen, Dana L. Penney, Randall Davis, David J. Libon, Rodney A. Swenson, Olusola Ajilore, Anand Kumar, and Melissa Lamar. Digital clock drawing: Differentiating "thinking" versus "doing" in younger and older adults with depression. *Journal of the International Neuropsychological Society*, 20 (9):920–928, 10 2014. doi: 10.1017/S1355617714000757.

[27] Hyungsin Kim, Young Suk Cho, and Ellen Yi-Luen Do. Computational clock drawing analysis for cognitive impairment screening. In *Proceedings of the Fifth International Conference on Tangible, Embedded, and Embodied Interaction*, TEI '11, pages 297–300, New York, NY, USA, 2011. ACM. ISBN 978-1-4503-0478-8. doi: 10.1145/1935701.1935768. URL `http://doi.acm.org/10.1145/1935701.1935768`.

[28] Randall Davis, David Libon, Rhoda Au, David Pitman, and Dana Penney. Think: Inferring cognitive status from subtle behaviors, 2014. URL `http://www.aaai.org/ocs/index.php/IAAI/IAAI14/paper/view/8626`.

[29] D. Shi, X. Zhao, G. Feng, B. Luo, J. Huang, and F. Tian. Integrating digital pen devices with traditional medical screening tool for cognitive assessment. In *2016 Sixth International Conference on Information Science and Technology (ICIST)*, pages 42–48, May 2016. doi: 10.1109/ICIST.2016.7483383.

[30] William Souillard-Mandar, Randall Davis, Cynthia Rudin, Rhoda Au, David J. Libon, Rodney Swenson, Catherine C. Price, Melissa Lamar, and Dana L. Penney. Learning classification models of cognitive conditions from subtle behaviors in the digital clock drawing test. *Machine Learning*, 102(3):

393–441, 2016. ISSN 1573-0565. doi: 10.1007/s10994-015-5529-5. URL http://dx.doi.org/10.1007/s10994-015-5529-5.

[31] Mihhail Lapuškin. Application for gesture based control of the pioneer robot with manipulator. Master's thesis, Tallinn University of Technology, 2012.

[32] Siim Kirme. Recognition of hand gestures using bezier curve and k-nearest neighbors method. Master's thesis, Tallinn University of Technology School of Information Technologies, Department of Computer Science, Chair of General Informatics, 2016.

[33] Samet Erap. Gesture based pc interface with kinect sensor. Master's thesis, Tallinn University of Technology, 2012.

[34] Kirill Buhhalko. Monitoring of the human motor functions rehabilitation by neural networks based system with kinect sensor. Master's thesis, Tallinn University of Technology School of Information Technologies, Department of Computer Science., 2013.

[35] Jevgeni Boruško. Alternative approach to model changes of human motor functions. Master's thesis, Tallinn University of Technology School of Information Technologies, Department of Computer Science., 2014.

[36] Helena Lissenko. Multi-kinect system for acquisition of turning motion. Master's thesis, Tallinn University of Technology School of Information Technologies, Department of Computer Science, 2015.

[37] Julia Koženkina. Quantitative analysis of the kinematic features for the luria's alternating series test. Master's thesis, Tallinn University of Technology School of Information Technologies, Department of Computer Science, Chair of General Informatics, 2016.

[38] W. Sui and D. Zhang. Four Methods for Roundness Evaluation. *Physics Procedia*, 24:2159–2164, 2012. doi: 10.1016/j.phpro.2012.02.317.

[39] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C. J. C. Burges, L. Bot-

tou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 1097–1105. Curran Associates, Inc., 2012.

[40] Yann LeCun and Corinna Cortes. MNIST handwritten digit database. 2010. URL `http://yann.lecun.com/exdb/mnist/`.

[41] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dan Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. URL `http://tensorflow.org/`. Software available from tensorflow.org.

[42] Vinod Nair and Geoffrey E. Hinton. Rectified linear units improve restricted boltzmann machines. In Johannes Fürnkranz and Thorsten Joachims, editors, *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, pages 807–814. Omnipress, 2010. URL `http://www.icml2010.org/papers/432.pdf`.

[43] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15:1929–1958, 2014. URL `http://jmlr.org/papers/v15/srivastava14a.html`.

[44] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2014. URL `http://arxiv.org/abs/1412.6980`.

[45] E. L. Lehmann and Joseph P. Romano. *Testing statistical hypotheses*. Springer Texts in Statistics. Springer, New York, third edition, 2005. ISBN 0-387-98864-5.

[46] B. L. WELCH. The generalization of 'student's' problem when several different population varlances are involved. *Biometrika*,

34(1-2):28, 1947. doi: 10.1093/biomet/34.1-2.28. URL + http://dx.doi.org/10.1093/biomet/34.1-2.28.

[47] Graeme D. Ruxton. The unequal variance t -test is an underused alternative to student's t -test and the mann–whitney u test. *Behavioral Ecology*, 17(4):688, 2006. doi: 10.1093/beheco/ark016. URL + http://dx.doi.org/10.1093/beheco/ark016.

[48] Jacob Benesty, Jingdong Chen, Yiteng Huang, and Israel Cohen. *Pearson Correlation Coefficient*. Springer Berlin Heidelberg, Berlin, Heidelberg, 2009. ISBN 978-3-642-00296-0.