TALLINN UNIVERSITY OF TECHNOLOGY
School of Information Technologies

Peeter Tarvas 206776IAIB

DRAWING STRATEGIES ANALYSIS FOR EMBEDDED FIGURE DRAWING TESTS

Bachelor's Thesis

Supervisor: Sven Nõmm
PhD

Co-supervisor: Aaro Toomela
PhD

Tallinn 2024

TALLINNA TEHNIKAÜLIKOOL
Infotehnoloogia teaduskond

Peeter Tarvas 206776IAIB

JOONISTAMISE STRATEEGIATE ANALÜÜS MANUSTATUD JOONISTUSTESTIDE JAOKS

Bakalaureusetöö

Juhendaja: Sven Nõmm
PhD

Kaasjuhendaja: Aaro Toomela
PhD

Tallinn 2024

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: Peeter Tarvas

16.01.2024

Abstract

Drawing Strategies Analysis for Embedded Figure Drawing Tests

This thesis analyzes multicube-type embedded figure drawing tests to help psychologists study links between education and cognitive ability. It studies drawing tests made by two different test groups to find segments and patterns that can be compared to each other. The second goal is to develop a methodology that can be applied to different subject groups and drawing tests. The final aim is to test if this data can be used in a wider scope such as making machine-learning algorithms.

In order to achieve these goals a workflow consisting of automatic segmentation based on pressure, object detection, ordered drawing sequence generation, pattern finding and data visualization is developed. To complete the second stage of these tasks a deep-learning computer vision model and a data mining technique are used to extract the data required and the author makes a set of methods to format the data into an explainable form that enables psychologists to analyze these subject groups. Also, a classification algorithm using the data in a formatted way shows that this method can be used to train and optimize prediction models.

The main results of this thesis are presented in the form of pattern and sequence heatmaps, a bar chart, and a line graph. A difference in the behavior of which two groups was observed in the patterns and sequences that they had used and how they had used them. There is also a demonstration of using this workflow to show that the results can be constructed on a different drawing test. The classification model is also tested and some optimization techniques that can be useful in the future are demonstrated.

The thesis is written in English and is 46 pages long, including 5 chapters, 18 figures and 3 tables.

Annotatsioon

Joonistamise strateegiate analüüs manustatud joonistustestide jaoks

Käesolev lõputöö analüüsib kuubiku tüüpu manustatud figuurjoonistusteste, et aidata psüholoogidel uurida seoseid hariduse ja kognitiivsete võimete vahel. Selle töö testrühmad jagunevad kaheks kelle joonistusteste uurides saab leida segmente ja mustreid. Teiseks eesmärgiks on arendada metoodika nii, et seda saaks rakendada veel teiste uurimisgruppide ja joonistustestide peal. Viimaseks eesmörgiks on testida kas arendatud andmete vormi saab rakendada laiemas ulatuses nagu näiteks masinõppe algoritmide peal kasutamisega.

Nende eesmärkide saavutamiseks töödati välja töövoog, milles osa koosneb rõhul põhinevast automaatsest segmenteerimisest, pilttuvastusest, õiges järjekorras joonise segmentide jadade genereerimisest, mustrite leidmisest ja andmete visualiseerimisest. Nende ülesannete käigus kasutati vajalike andmete saamiseks süvaõppe arvuti- nägemise mudelit, andmekaevet ning metoodikat, millega tulemusi vormistada lihtsalt tõlgendatavasse vormi. Samuti kasutati klasifitseerimise algoritmi kasutates andmeid, mis on saadud eelmistest tulemustest, et näidata kuidas saab masinõppe mudeleid treenida ja optimiseerida.

Lõputöö tulemused on esitatud mustrite ja segmentide soojuskaartidega ning tulpdiagrammi ja joondiagrammi kujul. Kahe rühma käitumise erinevust täheldati nende poolt kasutatud mustrite ja segmentide abil. Samuti näidati, et seda sama töövoogu saab kasutada ka teise joonise peal. Klasifitseerimise mudelit testiti samuti ning mõned optimiseerimise viisid on ka välja toodud, mis võivad tulevikus kasulikuks tulla.

Lõputöö on kirjutatud inglise keeles ning sisaldab teksti 46 leheküljel, 5 peatükki, 18 joonist, 3 tabelit.

List of Abbreviations and Terms

ASD	Archimedes Spiral Drawing
AI	Artificial Intelligence
CDT	Clock-Drawing Test
CNN	Convolutional Neural Network
CVAT	Computer Vision Annotation Tool
GSP	Generalized Sequence Pattern
GPU	Graphics Processing Unit
IDE	Integrated Development Environment
NMS	Non-Maximum Suppression
PDF	Portable Document Format
PNG	Portable Network Graphics
YOLO	You Only Look Once
YOLOv8	You Only Look Once version 8

Table of Contents

List of Figures

List of Tables

1. Introduction

The analysis of drawing strategies in embedded tests constitutes the scope of the present thesis. Although embedded tests were introduced nearly 100 years ago [1], [2] they did not receive as much attention as the Archimedes Spiral Drawing (ASD) test [3] or Luria's alternating series tests [4]. The goal of the embedded test is to assess the strategy (sequence of using different elements in drawing or tracing reference shapes) of the drawing process. The rationale behind this choice is that, unlike the simpler tests, it allows the freedom to choose in which sequence different elements of the reference figure will be drawn. At the same time, the reference figure has only 3 types of elements which simplifies the recognition task. The main novel component is the type of tests to be analyzed and the objective of the analysis. In the best knowledge of the author, AI-based approaches have not been used yet to extract and analyze drawing strategies. The latest positions this type of test between simpler tests like Archimedes' spiral drawing or Luria's alternating series and more complex tests such as Poppelreuter-Ghent's overlapping figures or clock-drawing test (CDT). During the test, the reference shape is shown to the patient, and then the patient is asked to trace (with a pen) the contours of the reference shape embedded in another figure. Compared to simpler tests, embedded tests provide more freedom to the patient, but are still more limited compared to Poppelreuter-Ghent or clock drawing tests.

The digitization of drawing tests started by [5] led to very precise results to support the diagnostic process. Statistical machine learning techniques were used successfully on a battery of tests by [6] later [7] demonstrated the necessity of supervised feature selection. For simpler tests, the main focus is mainly on the kinematic and pressure properties that describe the testing process. The ability of deep learning techniques to capture shape-related information has been acknowledged in [8] and some other articles. Although deep learning has also been shown to be a tool for research in the area of diagnostics [9] and [10]. Some results benefit from combining deep learning techniques as a drawing segmentation tool and statistical machine learning techniques as the classification tool working with kinematic and pressure features of the segments [11]. The latest work inspired current research. While the YOLO algorithm [12] is used to classify different image segments, the associative pattern mining algorithms [13], will be used to find strategies peculiar to different groups of subjects tested. It is important to note that the current thesis does not aim to support diagnostics but rather to provide machinery for academic research in the area of drawing test analysis.

Although digitization of drawing tests coupled with machine learning and artificial intelligence, after this shortened as AI, techniques resulted in highly accurate methods to support the diagnosis of neurologic diseases, the feedback from practitioners clearly points out the necessity of the tool to support medical and psychological research in the analysis of fine motor movements. Without such research developed methods, the risk remains in the areas of AI that it never will be used by practitioners on an everyday basis. Formally, the present work aims to provide the algorithm that segments the drawing into strokes and classifies each stroke to represent test drawing by the sequence elements. The working hypothesis stated by psychologists is that these sequences are expected to vary between subjects with different literacy levels. Researchers studying drawing tests from the psychological or medical side are interested in having such sequences not only to support diagnostics or find subjects with deviations in the education process but also to perform different studies uncovering the influence of education on motor skills.

The objective of this thesis is to construct sequences, patterns, and strategies from drawings depicting a multicube made by individuals from different backgrounds. These elements can help psychologists to better analyze the aforementioned study groups to find differences and similarities in their behavior and decision-making process. Moreover, the secondary aim of this project is to create methods that can be applied to analyze subject groups other than the one used in this paper. The final aim is to show that a machine-learning algorithm can be trained, optimized, and evaluated so that if in the future there is more data a prediction model can be made to classify individuals by their drawing pattern usage.

While previous research has delved into the analysis of drawing tests through the examination of semantic and kinematic parameters, as demonstrated in the work by [14], and has also investigated automated segmentation and analysis, as discussed in [15], the primary objective of this project is to establish a workflow and methodology tailored to enable psychologists to examine and compare strategies employed by distinct subject groups.

To complete this task a methodology solving image recognition, a way of describing a picture processable by algorithms and pattern finding must be developed. Even more, a very important task is to visualize the results in a way that psychologists can use easily to see the overall data that they want to analyze.

To solve these tasks, in this thesis, a workflow consisting of a computer vision algorithm, sequence generating method, a data mining technique, and data visualization libraries has been implemented.

The chapters of this paper consist of the following:

1. The rest of *Chapter 1* introduces the background that this project relies on and also formalizes the problem statement.

2. *Chapter 2* gives an overview of the data used in the project.

3. *Chapter 3* lists the tools used and also how the workflow is conducted in the implementation.

4. *Chapter 4* describes the results of the work and explains the meaning of them.

5. *Chapter 5* is the conclusion of the thesis.

1.1 Background

Education is a fundamental aspect of human development that shapes the way individuals perceive, interpret, and interact with the world around them. It plays a pivotal role in molding thinking patterns, which are integral to decision-making processes in various aspects of life, including those that involve cognitive abilities and creative expressions such as drawings.

As we navigate through the diverse educational landscapes and their impact on thought patterns, it becomes evident that education levels differ significantly among various populations. These differences can have profound implications on the cognitive patterns that shape our understanding of the world [16]. The influence of these thought patterns extends beyond academic learning and impacts the decisions we make, whether they involve complex problem-solving or the creative expression manifesting in our drawings.

The common ground between different levels of education, thought patterns, and cognitive disabilities is yet to be fully explored. For instance how the thought patterns differ from people who are illiterate or disabled to those who have at least basic education. Understanding the correlation between education and thought patterns can help with providing methods to discover or understand these people better.

While tablet and touch-screen technology is constantly evolving, numerous research studies present digitized versions of miscellaneous handwriting tests, that investigate drawings of circles, stars, spirals, and clocks. Others analyze sentences and character sequences.

The term strategies in this paper comes from psychology where it is defined as a plan designed to achieve a particular goal or target. These are the set of rules that are planned so that a person may achieve his goal without any trouble and with more facilitation [17]. In the context of this thesis, the term delineates the manner in which individuals execute their tests. The key attributes used to characterize these strategies encompass patterns, drawing strokes, their sequencing order, and the quantity of strokes required to finalize the image. Additionally, a feature of the strategy involves analyzing the distinctive patterns employed by each subject group.

Additionally, the administration of drawing tests in psychological or cognitive assessments has been made easier with the widespread use of tablets and touch-screen devices. The advent of digital handwriting tests, featuring various shapes such as cuboids, hexagons, or clock figures, enhances standardization and efficiency in data collection for research in these domains. Research studies have introduced digitized versions of diverse handwriting tests, ranging from the examination of drawings like pi-lambda lines or clocks, as demonstrated in works such as [15] [18], to the analysis of sentences and character sequences, as explored in studies like [19].

To shed light on the complex interplay of education, thought patterns, cognitive disabilities, and tablet-based assessments, we must also explore the technology that underpins the analysis of this intricate relationship. Computer vision, as a subfield of AI, is instrumental in the automated interpretation of visual information. In the context of this thesis, it serves as the key to unlocking insights from drawings and cognitive assessments conducted on tablets.

What's more, data mining is a process of understanding data by cleaning it, finding patterns, and creating models from the results. It is comprised of statistics, machine learning, and data management systems like databases [20]. In the context of this thesis data mining plays a central role in translating the data into a better format that can be used in the construction of the analysis.

Furthermore, the tasks of classification have been made easier with machine learning, which is also a subset of AI, by equipping us with tools to train automatic classifiers making it a valuable asset in analyzing for instance drawing and thought patterns that are from different individuals [21]. By employing machine learning classification techniques, we can discover patterns and anomalies in the cognitive responses recorded on tablets, shedding light on the impact of education and cognitive impairments. This novel approach offers the potential to find hidden insights, further enhancing our grasp of the intricate web of human cognition and educational influences.

In addition, classification techniques can be further used to train supervised machine-learning models such as Random Forests [22] or Decision Trees [23]. Moreover, an array of feature selection methods like Fisher score [24] and GridSearchCV [25] have been developed so that the training process can be optimized. This complimented with an evaluation method called Nested Cross-Validation [26] has made testing classification algorithms more accessible on smaller datasets.

In this thesis, all of the previously mentioned knowledge is applied by using methods already developed and also creating new methods to detect patterns and find strategies by analyzing multicube type drawings, that are made by people who have different educational backgrounds. These patterns can be found by using segmentation, classification, data mining, and data visualization techniques. The patterns can be used by psychologists to better understand the thinking process of people who have different levels of education. Moreover, the methods that are developed to complete this task can also be used to further analyze other types of drawings made by different study groups for example people who have different types of diseases that affect cognitive skills such as Parkinsons' disease. Finally, the pattern format can be used to train, optimize and evaluate a machine-learning model.

1.2 Problem statement

In this thesis, the primary focus of the investigation pertains to drawing patterns and strategies. Patterns are a set of commonly occurring sequences in a certain group of test subjects. Strategies on the other hand are the similar ways that the drawing process has been carried out in the test groups, this either being difference in pattern usage or in drawing sequences.

Embedded test is a drawing test that assesses the strategy and drawing process of the subject. The test involves drawing or tracing reference shape depicted for them to mimic. In the context of this thesis, the tests were carried out on two groups the literate who have received an education and the illiterate who have not.

The present research applies a deep neural network classifier to drawing tests to find ordered segments and use those to discover common drawing patterns used by the subjects who have drawn the figures. From these patterns, segments and data gathered from the drawing process, strategies can be deducted that can be used to compare and analyze the testing groups by psychologists. Moreover, the second aim of this thesis is to develop the methodology in such a way that it could be used on similar drawing tests on other test groups. It is also shown that with this workflow and data representation, a classification algorithm can be trained and optimized.

To achieve this objective, several challenges must be addressed:

1. Develop a method for determining the segments with the correct drawing order within a closed figure.

2. Classify each segment.

3. Devise a mechanism to investigate the most common subsequences of strokes to compare the two groups with different educational backgrounds.

4. Create a visualization strategy for presenting both the intermediate and final results in a format applicable to analysis.

5. Find a way to train and optimize a classification algorithm.

2. Data

All test data was in JSON format and contained recorded data of the drawing tests, which included metadata (session id, test type, anonymous identification number) and an array of data records containing dynamic features. The following dynamic features (timesequences) were captured by the tablet: X-coordinate (mm); Y-coordinate (mm); timestamp (sec); pressure (arbitrary unit of force applied on the surface: [0,..., 6.0]); altitude (pen inclination, rad); azimuth (pen orientation, rad). It is also apparent that each subarray was one stroke of the entire picture because either the starting or finishing element always had the pressure parameter as 0. JSON format that can be turned into drawings is more useful since the data in its raw format can be easily used to segment different lines.

2.1 Drawing tests

Embedded drawing tests are tests where the test subject has to find or copy a certain object that has been shown to them.

2.1.1 Multicube drawing tests

Multicube drawing test is used to measure visuospatial ability and to a lesser extent measure constructional praxis. In the multicube test, the person drawing has to find and mimic the top example cube in the bottom object.

Here is the test that the participants have to complete:

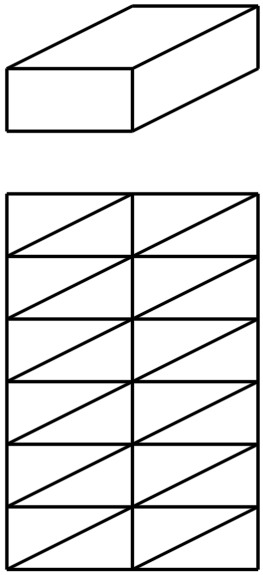

Figure 1. Multicube test reference shape

2.1.2 Hexagon drawing test

The other test considered for this thesis is a hexagon drawing test. In this task, the person who is drawing has to copy the object that is presented to them in the test on a white page.

The following figure shows the hexagon object that the test subjects had to copy:

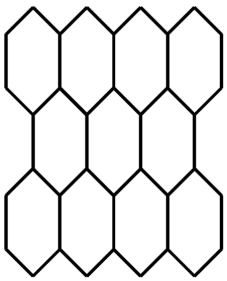

Figure 2. Multicube test example

It was decided that the main figure to be researched in this thesis would be the multicube. Even though the author didn't include the hexagon figure as the main test to be researched it was used to show that the methodology can be applied to other object types besides the multicube.

2.1.3 Training and validation data

The author decided that the training of a YOLOv8 algorithm that can detect three different shapes and the testing data was limited the author created some data for training manually. The author drew 18 images in extra from which he could get segment images for detection training, there were 6 hexagons, 6 multicubes, and 6 line images drawn. The line images are images where the author drew just the 3 different shapes as much as possible so that there would be enough data to train on. In the end, 609 different segments for training and validation were created of which 36 were for validation and 573 for training.

2.1.4 Test data

There were 70 multicube drawing tests in total gathered from the subjects with different levels of literacy. The distribution of the patients by education level is the following:

- 18 literate
- 52 illiterate

Since the distribution of data is very uneven modifications had to be done in order to work with it. We have to balance the data so that the classes have the same amount of subjects because otherwise, the results could not be analyzed by comparing the two groups, this resulted in the following distribution:

- Literate - 18 individuals
- Illiterate - 18 individuals

2.1.5 Note about data acquisition

The data acquisition, anonymization and processing were conducted in accordance with the privacy-preserving laws and with the permission of the ethics committee. Principal investigator: Aaro Toomela Research project name in Estonian: Kultuuri-, bioloogiliste ja arenguliste tegurite roll kognitiivse reservi mehhanismides ja kognitiivse taandarengu ennetamises Taotlus nr 6-5.1/14 Permitted by: Ethics committee of Tallinn University 12. mai 2021 decision nr 12.

3. Implementation

The workflow was done with Python(3.11) [27] programming language in PyCharm IDE. Author used the Conda package manager [28] and Jupiter Notebook [29]. In addition, the following libraries, GitHub open-source projects, and a online tool were used:

3.1 Overview

1. Matplotlib – a Python library used for visualization of data. This was the main library used for generating and enhancing images [30].

2. NumPy and Pandas - open-source libraries for more effective computation in Python [31], [32]. These libraries were used for data storage, manipulation and overall utility

3. Ultralytics - a library that provides state-of-the-art AI tools for image detection and classification like YoloV8 model [12].

4. PyTorch - is a machine learning framework based on the Torch library, used for applications such as computer vision and natural language processing [33].

5. Seaborn - is a Python data visualization library based on Matplotlib. It provides a high-level interface for drawing attractive and informative statistical graphics [34].

6. Gsppy - a project on GitHub that has an easy-to-use Generalized Sequence Pattern used for sequence data mining [35].

7. NMS-python - a project on GitHub that provides a good Non-Maximum Suppression algorithm for bounding-box selection [36].

8. CVAT.ai - is an online tool that is used for picture annotation [37].

9. Scikit-learn - is a Python library used for Machine Learning algorithms in the classification step [38].

3.2 Descriptions

To have a better understanding of the technologies and algorithms used here are brief descriptions of the methods used.

3.2.1 Pytorch

Pytorch is an open-source machine learning framework based on the Python programming language and the Torch library. Torch is an open-source machine-learning library used for creating deep neural networks and is written in the Lua scripting language [39]. In the context of this paper, the framework was used to support GPU-based image model training. TensorFlow was also considered as an alternative but because Pytorch is easier and lighter to work with the latter option was picked.

3.2.2 Convolutional Neural Networks

Convolutional Neural Network or CNN for short is a type of Deep Learning neural network architecture commonly used in images, audio, and text for classification. Moreover, it works by predominantly extracting features from a grid-like matrix dataset, where for example data patterns play an extensive role. A CNN consists of multiple layers like the input layer, convolutional layer, pooling layer, and fully connected layers. The convolutional layer applies filters to the input image to extract features, the pooling layer downsamples the image to reduce computation, and the fully connected layer makes the final prediction. The network learns the optimal filters through backpropagation and gradient descent. The output from the fully connected layers is then fed into a logistic function for classification tasks like sigmoid or softmax which converts the output of each class into the probability score of each class [40].

3.2.3 YOLOv8

You Only Look Once in short abbreviated YOLO, also in the context of different versions shortened as YOLOvX for example YOLOv8, is a state-of-the-art computer vision model built by Ultralytics. The model can do object detection, classification, and segmentation tasks which are all accessible through a Python package. The algorithm uses end-to-end neural networks that make predictions of bounding boxes and class probabilities all at once. While other Convolutional Neural Networks work by detecting possible regions of interest using the Region Proposal Network and then performing recognition on those regions separately, YOLO performs all of its predictions with the help of a single fully connected layer. YOLO also only performs a single iteration over the image making it a lot faster than the alternatives. The model has had many versions from YOLOv2 to YOLOv7 but in this project, the newest version YOLOv8 was used because of improved developer experience, like being able to install the library through a package manager [12].

3.2.4 Data annotation

Data annotation is the process of labeling data with relevant descriptions to make it easier for computers to understand and interpret. Usually, data annotation is done manually but as machine learning algorithms advance the procedure is becoming more and more automated. Because there isn't a public data set that has the necessary objects required to train a computer vision object detection model needed for this project, the author made one with the data annotation tool Computer Vision Annotation Tool or for short CVAT.ai [37]. It's a free, open-source digital image annotation tool that has all the methods to make the data set the training process needed.

3.2.5 Generalized Sequential Pattern algorithm

Generalized Sequential Pattern(GSP) algorithm is used for data sequence mining. Usually, it is used for databases but it can be used also for other applications. The algorithm uses apriori on its base level to find commonly occurring items. It starts with finding the frequent items of size one and then passes that as input to the next iteration of the GSP algorithm. The database is passed multiple times to this algorithm. In each iteration, GSP removes all the non-frequent itemsets. This is done based on a threshold frequency which is called support. Only those itemsets are kept whose frequency is greater than the support count. After the first pass, GSP finds all the frequent sequences of length-1 which are called 1-sequences. This makes the input to the next pass, it is the candidate for 2-sequences. At the end of this pass, GSP generates all frequent 2-sequences, which makes the input for candidate 3-sequences. The algorithm is recursively called until no more frequent itemsets are found or the maximum item size is reached [41].

3.2.6 Non-Maximum Suppression algorithm

Non-maximum suppression(NMS) is a post-processing technique that is commonly used in object detection tasks to eliminate duplicate detections and select the most relevant bounding boxes that correspond to the detected objects. It is a critical step in many computer vision applications such as face detection, pedestrian detection, and object recognition. In the context of this paper, it is used after YOLOv8 object detection. This is because picture segments can get two or more bounding boxes even though they only should have one [42]. This algorithm has an open-source implementation available which was used in the development of the methods [35].

3.2.7 Data visualisation

Matplotlib is a Python library used for creating visualizations. The main feature of the extension is the ability to plot graphs with x and y parameters. This feature enables doing a lot of the necessary tasks like plotting a multicube-shaped object from the JSON file or its individual elements. It also enables saving the plots in PNG, PDF and other formats which makes it ideal for creating images [30]. Seaborn is an extension of Matplotlib. It adds more useful and easy-to-use methods that are common in data visualization. For example, the library adds: Relational plots, Distribution plots, Categorical plots and many more useful plots that make work more efficient when dealing with more complex data-related tasks [34]. This project requires that a lot of different styled plots be used in order to show the final results in a simple and meaningful way and that is why this library is used.

3.2.8 Classification

Classification is a supervised machine-learning process where input data is categorized into classes based on one or multiple features. It can be performed in structured and unstructured data to predict if the data will fall into predetermined categories. It generates a probability score for the data so that it can be determined if the label is correct or not. Some applications for machine learning would be Image classification, Medical diagnostic tests or Malware classification. Classification models are trained on a pre-labeled dataset from which the probabilities for predictions are concluded from [43].

3.2.9 Decision Tree

Decision Tree is a supervised machine-learning algorithm used for both classification and regression problems. It uses a flowchart-like tree structure where each node is a feature, branches denote the rules and leaf nodes give the results [23]. In this thesis, a Decision Tree implementation from the Scikit-learn library is used [38].

3.2.10 Random Forests

Random Forests is a supervised machine-learning algorithm that uses a combination of tree predictors such as Decision Trees to classify input data. The method depends on the value of a random vector sampled independently and with the same distribution of all the trees in the forest. Random Forests can be tuned to use different amounts of trees to decrease the error rate that a single tree would have [22]. Scikit-learn [38] has an out-of-the-box implementation for this algorithm which is used in this project.

3.2.11 Supervised Feature Selection

Large datasets can have a lot of features to train a classification model on. This can become a problem since only some parameters can give a better prediction than others for a specific training set. Supervised Feature Selection can improve this process by picking out the best descriptors for the specific target that needs to be predicted [44].

3.2.12 Cross-Validation

Cross-validation is a statistical method used to estimate the skill of a machine-learning model. It is commonly used when a limited data sample is only available by splitting the data into different folds that are used as training sets and a test set and then evaluating the model design. This in turn can be used to determine if the data is suitable for making a prediction model or not [45]. In this thesis, the StratifiedKFold algorithm from Scikit-learn [38] was used which can shuffle randomly and divide the data so that each fold has an equal number of classes that in turn will decrease fold bias [46].

3.2.13 Fisher Score

Fisher Score is a supervised feature selection method that finds which parameters are the most important in the group. It selects each feature independently according to their scores under the Fisher criterion which leads to a suboptimal subset of features. Fisher's score uses a ratio of between-class variance to within-class variance. [24].

3.2.14 GridSearchCV

GridSeachCV is a tool for fine-tuning the parameters for a machine-learning model such as Random Forests. It generates combinations of all the specified candidate features from a grid parameter value that is given an input. The model and data used are specified beforehand as variables that the search uses to then evaluate with cross-validation. There is a downside to GridSeach in that if it is used with cross-validation it takes a lot of time to cumulatively evaluate the best parameters [25]. In this project Scikit-learn was used to apply GridSeachCV [38].

3.2.15 Nested Cross-Validation

Nested Cross-Validation is an approach to model hyperparameter optimization and model selection that attempts to solve the problem of overfitting the dataset [26]. The method uses an inner validation split which means has a double-loop consisting of an outer loop, that will serve for assessing the quality of the model, and an inner loop that provides model or parameter selection. Both of the loops are independent so one step of cross-validation does only one thing [47].

3.3 Workflow overview

Since in the project, a computer vision model needed to be trained the workflow is divided into two different step-by-step working orders. The first workflow is about YOLOv8 model training with a custom data set. The second one is about the pattern and strategy finding.

3.3.1 Workflow for training

1. Data pre-processing and segmentation

    (a) Training and validation data generation. Using the same type of tablet that the data was gathered with, draw different types of lines that are the main classifiers for the pictures.

    (b) Creation of segments from the whole picture that is formatted as a JSON file so that we have a sequence of lines that make up the whole drawing.

    (c) For the training and validation data, the pictures are annotated using the CVAT.ai tool to make them usable with the YoloV8 detection algorithm.

    (d) Training pictures and labels are separated into different folders so that they can be used for the YOLOv8 algorithm correctly.

2. Training server setup

   (a) Because the YOLO algorithm requires a lot of resources, TalTechs AI-Lab server, which has the GPU power to better train computer bision models, is used.

   (b) Set up all of the required packages with the Conda package manager so that a server can be used to train the YOLOv8 model.

3. Detection training

   (a) Using the YOLOv8 algorithm to train a detection model for the different shapes being used as classifiers.

3.3.2 Workflow for pattern finding and classification tests

The stages for finding patterns are the following:

1. Data pre-processing and segmentation

    (a) Dividing data into 2 separate groups.

    (b) Randomly balancing testing data so that the size of both of the classes is equal.

    (c) Creation of segments from the whole picture that is formatted as a JSON file so that we have a sequence of lines that make up the whole drawing.

    (d) All of the testing data, that is used to find patterns, is structured in the directories so that the test subjects' data is in one folder that contains the sequence of pictures that are ordered by the drawing order.

2. Using detection to classify pictures

    (a) Using Non-Max Suppression to combine overlapping similar bounding boxes that have been detected.

    (b) Generate sequence patterns that correlate to the sequences that are detected in the picture.

    (c) Order sequence so that the line drawing order is correct.

3. Pattern mining

    (a) Format sequences in a comma-separated file-like format so that they could be used for sequential pattern mining.

    (b) Use the Generalized Sequence Pattern mining algorithm to find patterns.

4. Pattern matching

    (a) From the sequences generated find corresponding patterns for each drawing sequence.

    (b) Match the corresponding subject identifier and sequence to the array of patterns that they have used.

5. Result generation and visualization

    (a) Heatmaps

        i. Generate heatmaps for pattern distribution throughout drawing iterations.

        ii. Generate heatmaps for segment distribution throughout drawing iterations.

    (b) Generate bar charts to show pattern frequency between the different study groups.

    (c) Generate a filled line graph that shows how many people are drawing each segment drawing iteration from different study groups.

6. Classification testing

    (a) Convert the patterns to a frequency table where each subject pattern usage and class is represented.

    (b) Test the patterns as features with Nested Cross-Validation to show that the data can be used for classification.

        i. Use Fisher Score to find good patterns that predict if the person is educated or not by testing the subfeatures with a Decision Tree.

        ii. Use Random Forests with GridSeachCV to find the number of trees that provide good results.

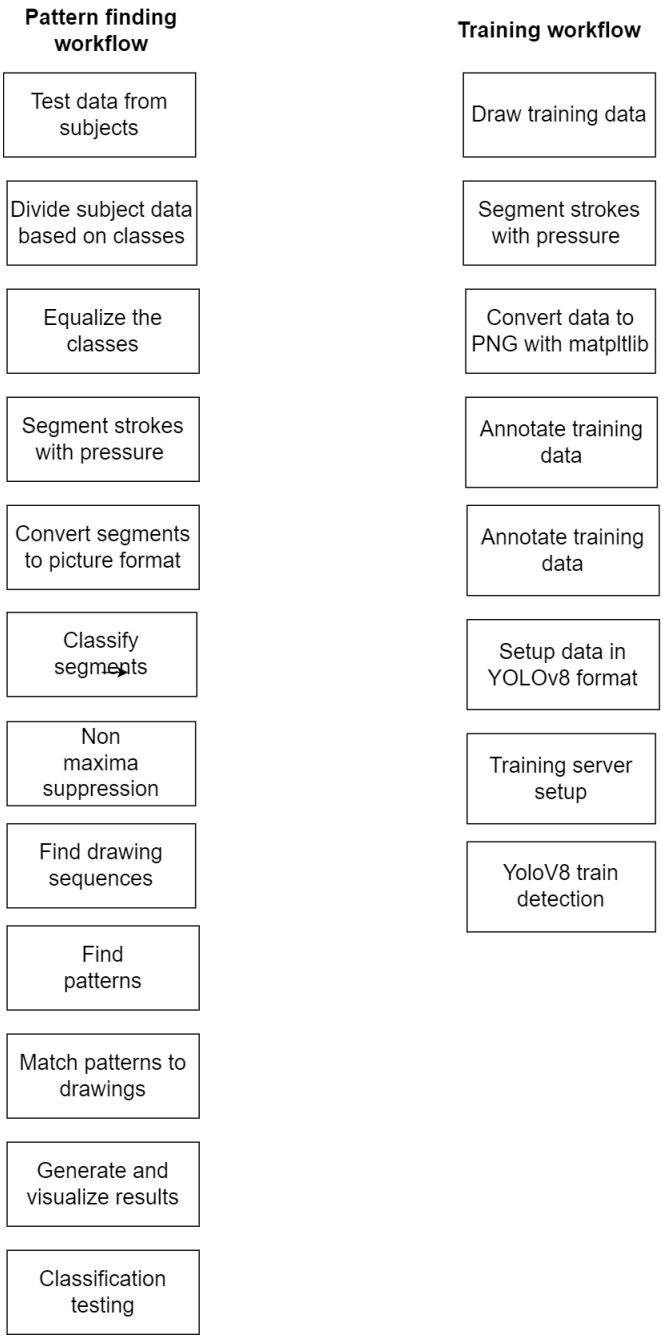

Figure 3. Workflow

4. Results

This chapter gives the descriptions of the results from the implementation of the algorithms and discusses them. It also shows that the workflow can be used on other drawing tests and that the data formatting can be used for training a classification algorithm.

4.1 Segmentation by pressure

From the pictures that were generated, the author came to the conclusion that just a simple classification of the whole image would not work because the pictures varied in quality, and having all the possible variations of drawings to train a computer vision model is nearly impossible. To deal with this problem a detection model that recognizes separate segments of the whole drawing would be a better classifier because with it multiple objects can be detected from a single image while the main shapes that have to be detected would be simple.

Three different segment lines were chosen by which objects in the picture can be discovered and classified with:

1. Horizontal line

2. Vertical line

3. Diagonal line

With these three segment types, all of the pictures can be classified. Furthermore, these three elements make it easier to describe drawing sequences in a text format which is important in the next steps.

Due to the fact that segment order is important in the results then the proper ordering of the drawing must be maintained throughout the object detection part of the project. This is done by not detecting objects from the drawing as a whole but splitting the picture up into many smaller pictures that have a segment that is drawn as one stroke. A stroke in this context means that a line is drawn without lifting the pencil tip from the tablet. The drawing can be easily divided into many smaller pictures because it's in JSON format where each drawing point has pressure as a parameter and each stroke is a sublist. This

way of partitioning the picture has two positive effects:

- Objects can be detected more easily by dividing them into smaller parts.
- The order of the strokes is preserved during detection

All of the pictures were split into multiple smaller pictures by pressure. Each drawing has its own directory where all the smaller pictures are stored and the pictures are enumerated by drawing order.

4.2 Yolo tuning

Because there aren't any public models with the kind of objects this work requires a custom YOLOv8 model to be tuned in a way that the three different segment types can be detected.. The training data was generated manually by the author and in the end, it consisted of 609 different annotated pictures for training and validation of which 36 were for validation and 573 for training. The annotation of the pictures was done manually as well using the CVAT.ai tool [37].

After the data was gathered it was formatted to be used by the YOLOv8 training algorithm. It consists of separating the validation and training data into separate folders and also separating the labels from the pictures.

YOLO also has an in-built picture augmenter which was used. After testing with different parameters the optimal way of training was the following:

- degrees=5.0
- translate=0.2
- perspective=0.00001
- scale=0.5
- shear=0.1
- mosaic=0.2
- mixup=0.2

During the training of the YOLO detection model, 268 layers were created and a total of 43 608 921 parameters were used. The whole process was 400 epochs - iterations - and lasted a bit over 2.5 hours. The overall results are in the following table:

Precision tells the accuracy of the detected objects, indicating how many decisions were correct. Recall denotes the ability of the model to identify all instances of objects in the images.

Table 1. YOLOv8 training performance metrics

Results		
Class	Precision	Recall
all	0.826	0.974
vertical	0.988	1
horizontal	0.603	0.923
diagonal	0.886	1

Considering the limited amount of training data that is available and also that the YOLO algorithm made augmentations to the pictures to widen the variety of data the results were considered to be acceptable. Moreover, while it is true that for instance, horizontal segment training could be better the main motivation of this thesis is not to train the most precise Computer Vision model but to use that model in other tasks.

4.3 Object detection

After the model is trained and each picture is split by drawing pressure into multiple pictures the object detection can be applied.

To begin with, during the object detection, there is a common error where one object had multiple bounding boxes. This is due to the fact that the computer vision model has not been trained perfectly but as a fix to this problem the author is using the Non-Maximum Suppression algorithm which eliminates duplicate detections and selects the most relevant bounding box to represent the object. Since the bounding box size precision isn't important, but just the classification of what is on the picture then this method works for this task. The maximum overlap that the objects can have is 20%. Also, 5%, 10%, and 50% are tested but they don't have as good of a result as the 20% overlap.

Figure 4 demonstrates the object detection. The upper left picture is the detection before the NMS and the bottom one is after the algorithm has been used. The picture on the right shows where the line resides in the picture as a whole.

It can be seen that before the duplicate elimination, there are 2 bounding boxes over the single horizontal line. After the algorithm, only one remained.

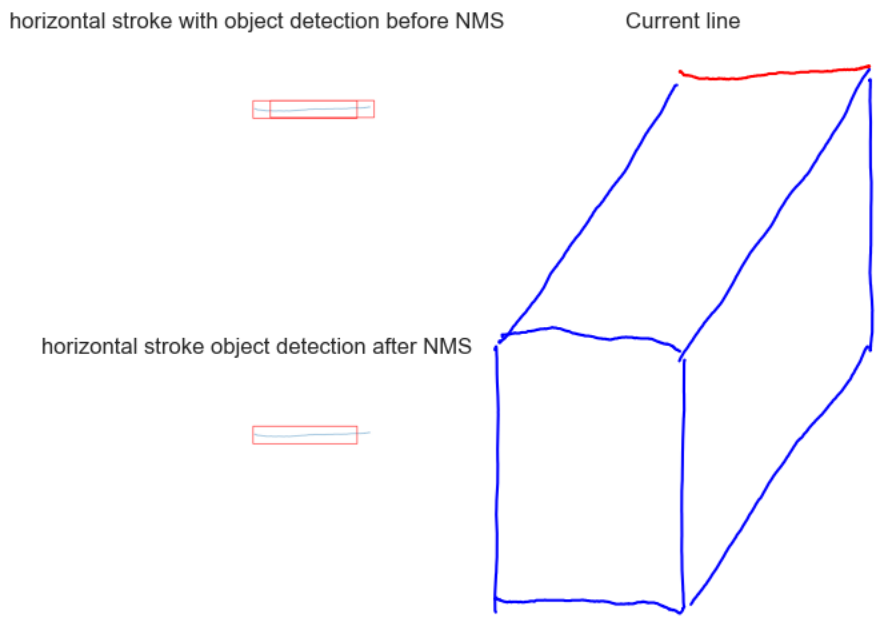

Figure 4. Multicube segment Non-Maximum Suppression(NMS)

4.4 Sequences

A drawing sequence is the order and type of strokes with which the picture is made with.

The sequences were generated by:

- Classifying objects in the segmented pictures that belong to a drawing.
- The segments from classified pictures are converted into sequences in the format where:
  - ’d’ - diagonal
  - ’h’ - horizontal
  - ’v’ - vertical
- Lists of these elements from the picture that they belong to are then added to the sequence that is ordered starting from the first picture up to the last picture that belongs to the drawing.

- All of these sequences from one education category are then added to a nested list where each sublist is the drawing sequence of one whole picture.

Example of one drawing sequence that was generated by this method:

'd', 'h', 'v', 'd', 'd', 'h', 'v', 'v', 'h'

4.4.1 Heatmap

From these sequences, it is already possible to find most popular drawing segments used. Each individual has a certain drawing order which may be the same with other people's drawing orders. The following picture shows the segment normalized frequency overlap between the illiterate class of test subjects. The Y-axis is the drawing iteration and the X-axis has the three drawing segments: d, h, v. The left picture has the normalized frequency of all the strokes made in each iteration, where the darker the color higher the frequency as can be seen from the color bar next to the heatmap, and the right figure shows the most popular segment drawn a each certain drawing moment.

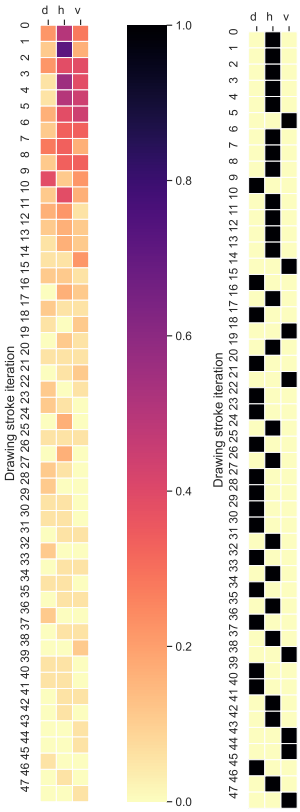

Figure 5. Multicube illiterate sequences normalized and most popular

For comparison here is the literate dataset sequence heatmaps that are generated with the same methods. As can be seen, they differ from each other when it comes to the most popular segment used in each drawing moment.

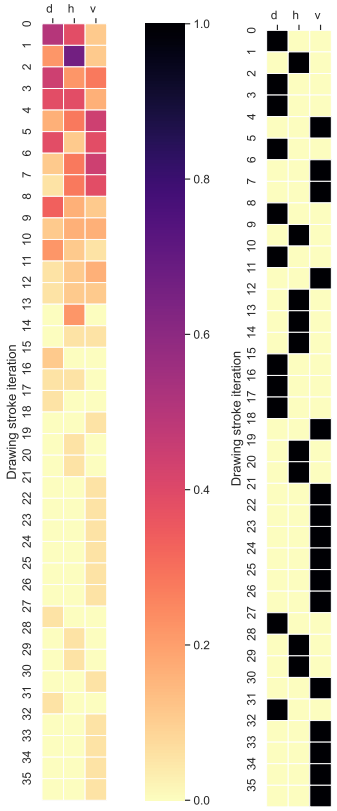

Figure 6. Multicube literate sequences normalized and most popular

Between the two classes, the main noticeable difference is that illiterate subjects use the horizontal line a lot more, that being 23 out of 47, often than the literate where the sequence occurred as the most popular one 9 times out of 36. Moreover, the literate tend to be more spread out with their most popular segment of drawing initially where the distribution of the most popular segments is the following for the first 12 rows: 6 diagonal, 2 horizontal and 4 vertical. The reverse is true for the illiterate who tend to focus more on drawing horizontal lines where in the first 12 iterations most popular segments were: 1 diagonal, 10 horizontal and 1 vertical.

4.4.2 Drawing count

The difference in drawing strategies can also be shown by the number of people who are drawing at one moment. The following graph shows how many people are drawing at one stroke moment in each class. The X-axis is the current drawing iteration and the y-axis is the number of people drawing at one moment. In the graph, the illiterate and literate are marked as green and blue respectively, their overlap has a dark turquoise-like color.

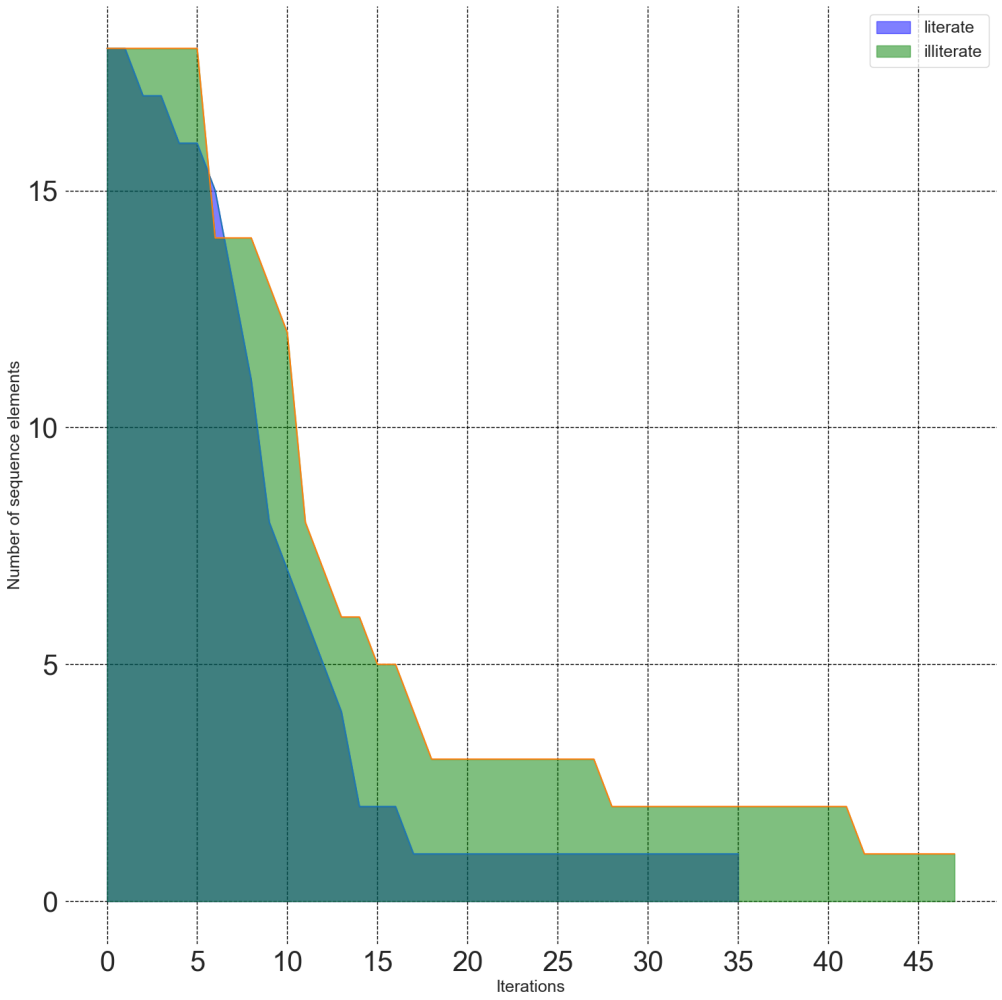

Figure 7. Multicube Drawing Iterations Graph

As can be seen, most of the literate study group has finished around the same stroke drawing moment interval which is between the 4th and 14th iterations. On the other hand, the illiterate group had a similar finishing curb but a bit later starting around 8th and ending at 15th segment drawing moment, and overall they had more variety in finishing moments because their graphs decline starts to even out around the 17th iteration. Moreover, the illiterate group tends to make more drawing strokes than the literate because the iterations where the number of drawers in the educated group is higher than the uneducated is only

between the 5th and 10th x-axis points whereas in the rest of the graph, the latter group has always more participants then the former one.

4.5 Patterns

For a better and more detailed analysis patterns are found that emerge from the sequences of the two different groups.

4.5.1 Generalized Sequence Pattern

Utilizing the Generalized Sequential Pattern Mining algorithm, which has an open-source implementation in Python, specifically the gsppy library by Prado Lima, was employed [35]. The algorithm was executed on distinct datasets representing literate and illiterate groups independently. The primary objective was the identification of patterns capable of distinguishing between these two observable groups.

Experimentation was conducted by varying the sizes of patterns and exploring configurations with 3, 4, and 5 items per pattern. Additionally, different support, ranging from 0.6 to 0.2, were examined. Following this iterative process, the optimal configuration was determined to be patterns comprising 3 items each, with a support of 0.3.

After setting the optimal parameters for data mining the patterns were found and matched to all of the sequences where they occur and in their occurrence order. Moreover, overlapping pattern instances are also included in this. This way all of the patterns used by each individual were found and a comparison between the two study groups could be made by analyzing the patterns those groups use.

At this point, it should be addressed that there is a problem with this approach which is that the results of the algorithm cannot be validated. Apriori algorithm was tested to verify the results but this turned out not to work because of the data structure the sequences were in. The author looked into other ways to validate the algorithm but there weren't any good solutions to be found. Even though this problem could not be solved it was decided that the results would be used to finish the project.

4.5.2 Bar chart

For comparing the two classes a bar chart is generated where each data point on the horizontal is a pattern and on the vertical axis are the number of occurrences ranging from 0 to 25. The illiterate group is marked as the blue bar and the literate as the orange bar.

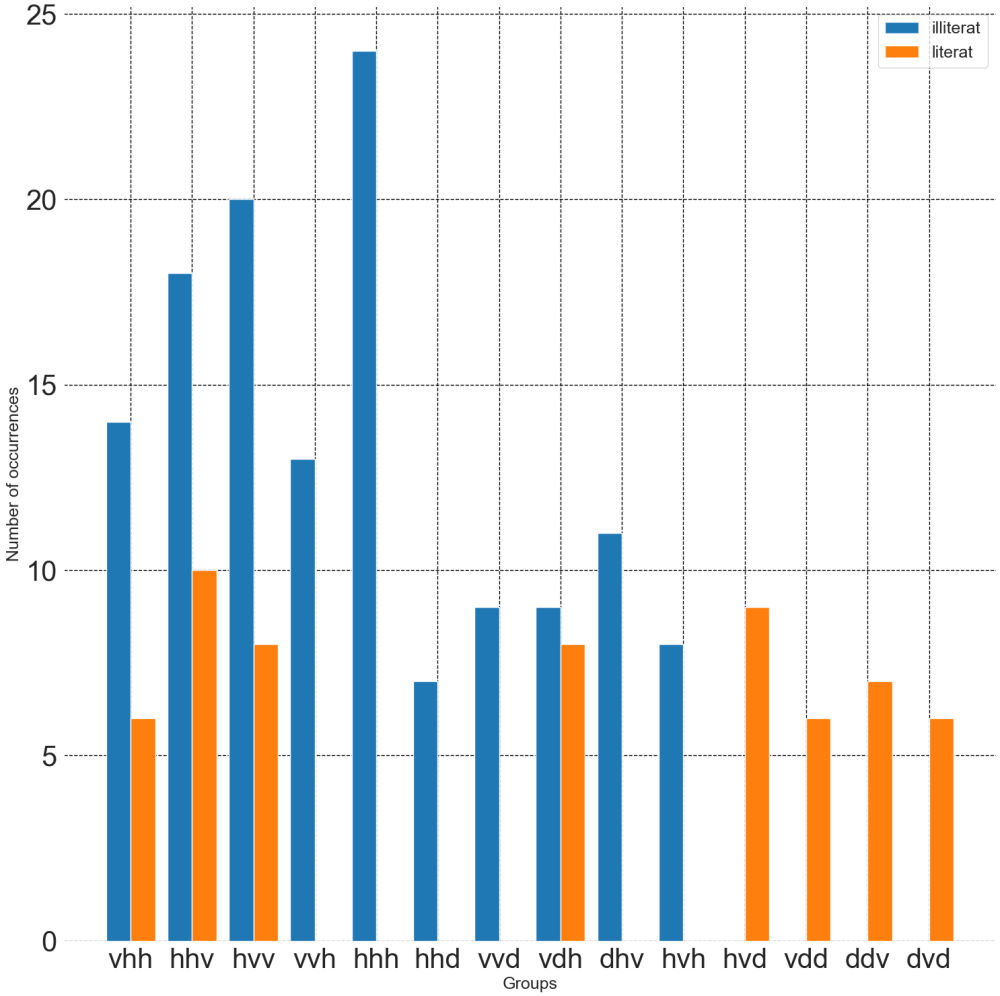

Figure 8. Multicube Pattern Occurrences Bar Chart

From this figure, a comparison between the two groups can be made. Firstly, only 4 pattern types overlap between the two groups. Here is a list of overlapping and individual patterns:

1. Overlapping: vhh, hhv, hvv, vdh

2. Illiterate: vvh, hhh, hhd, vvd, dvh, hvh

3. Literate: hvd, vdd, ddv, dvd

Furthermore, It is apparent that the illiterate group has used a bigger variety of patterns and that the usage of these patterns has happened more often than in the literate group. This can be seen as there are 6 elements where the count is over 10 in the illiterate group whereas in the literate group, there are none. The explanation for this can be derived from the fact that the illiterate group has made more strokes overall while drawing the picture and because of that the patterns also occur more.

Moreover, in the literate section, the patterns have a more equal distribution where all the pattern usages are between 5 and 10. While in the illiterate part, certain elements, hhh, hvv, hhv, have occurred more than 15 times, vhh, vvh, dvh more than 10 times and hdd, vdd, vdh, hvh less than 10 times.

4.5.3 Heatmap

To conduct more analysis a heatmap similar to the sequences one is used to show pattern distribution over time and to discover if there are any differences between the two groups. On the X-axis are the different patterns that the groups have used. These patterns can differ by class as seen in the bar chart before. The Y-axis represents each drawing moment that a stroke has been made. Because different people require a diverse number of strokes to make one drawing then the Y-axis length can also differ from class to class.

In the following images, the frequency at which each pattern appears is normalized to the number of patterns drawn initially. In this way, we will see that there are higher frequencies at the beginning of the heatmap but in the end, there aren't that many used patterns anymore. This is because fewer people are using common patterns at these moments.

The right image in Figures 10 and 6 show the most popular pattern in each drawing iteration by representing them in black squares.

Firstly, the illiterate group had the following patterns in their heatmap: dvh, hhd, hhh, hhv, hvh, hvv, vdh, vhh, vvd and vvh an the maximum amount of patterns that someone applied was 19.

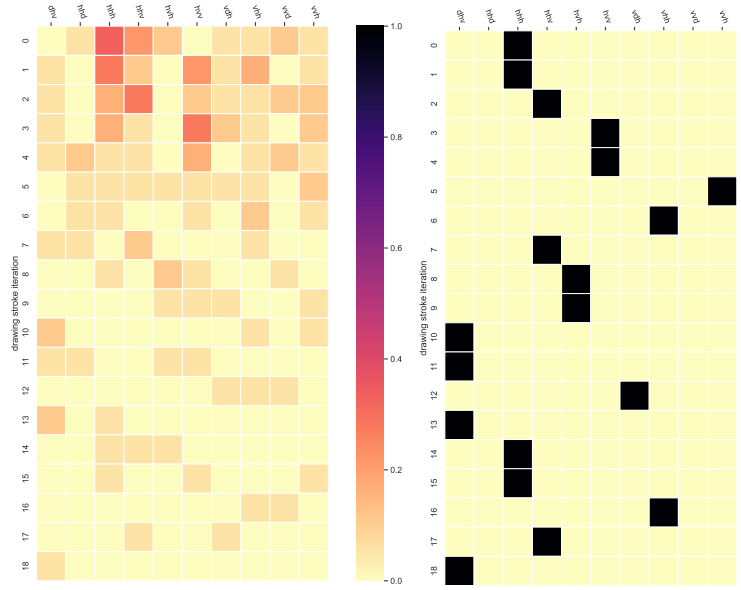

Figure 9. Multicube illiterate patterns normalized and most popular

It can be seen that in the beginning, the patterns that start with h: hhd, hhh, hhv and hvh, except for huv, are more popular than the other ones. This is also apparent from the second image where initially all of the most popular patterns start with the horizontal line. After the 6th drawing moment, the pattern slots usage starts to decline in the sense that on the 7th iteration, 4 boxes are only over 0.0 which means that 60% of the patterns aren't used in each iteration from that point onward, this being either because the subjects have stopped drawing or they are not using popular patterns anymore.

The literate group on the other hand had the following patterns: ddv, dvd, hhv, hvd, hvv, vdd, vdh and vhh. Their heatmaps were constructed the same way as the illiterate ones. The maximum amount of patterns that had been made was 12.

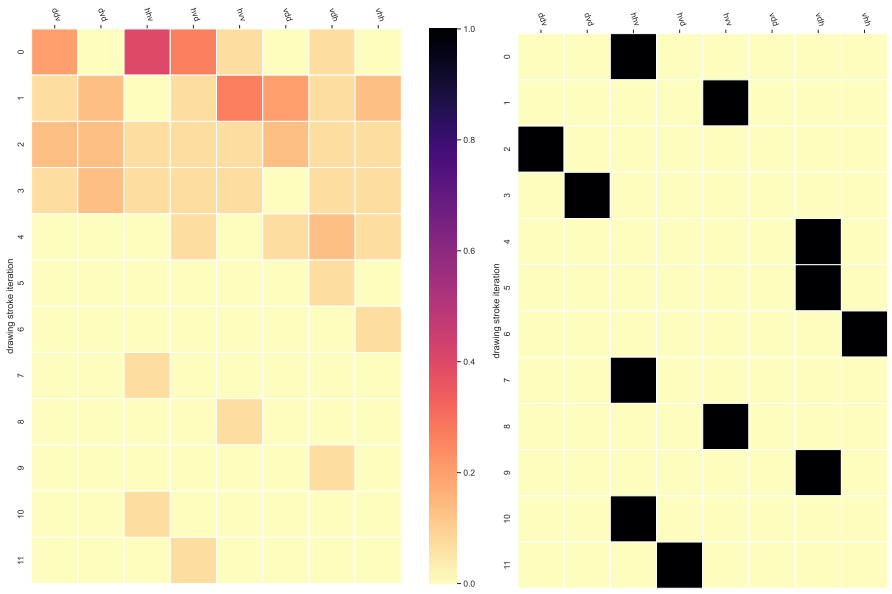

Figure 10. Multicube literate patterns normalized and most popular

The most popular pattern and iteration was hhv in the first row. Moreover, patterns that start with a horizontal line seem to be the most popular initially, followed by patterns starting with the diagonal line. There seem to be fewer patterns used overall because, after the 4th pattern drawing moment, there is only 1 slot out of 8 used till the end, meaning that 12.5% of the boxes were used after that. This can be either because they have stopped drawing or because they do not use a common pattern anymore. By looking at the sequences in Figure 7 it can be concluded that only 1 test subject didn't finish their drawing after the 20th stroke. Due to this, it can be concluded that there are less patterns overall because all but 1 of the individuals have finished their test after the fourth or 4th pattern drawing moment.

4.6 Method on different data

To show that these methods can be applied to more shapes than the multicube the same methodology is applied for showcase sake to the hexagon shape with 2 groups of observables.

4.6.1 Object detection

Here is a visual example of a drawing line being extracted from the hexagon test and then it is detected with YOLOv8.

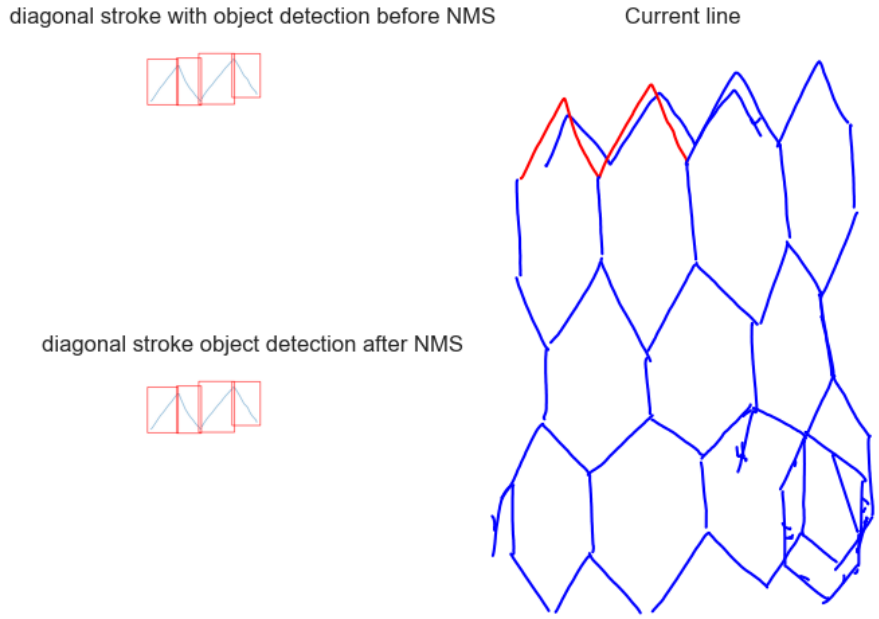

Figure 11. Hexagon segment with Non-Maximum Suppression(NMS)

4.6.2 Sequences

The sequences detection worked the same way as it did in the section *4.4* and after applying the same methods that were used in the subchapter. The results were the following.

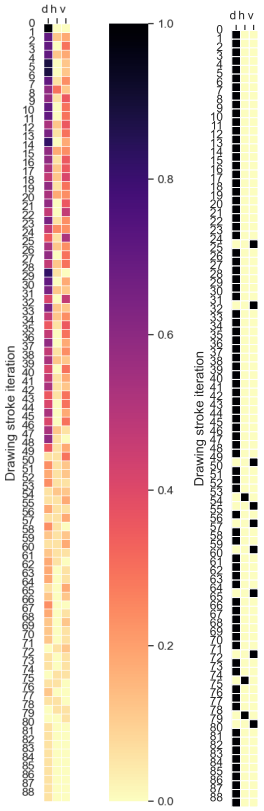

Figure 12. Hexagon literate sequences normalized and most popular

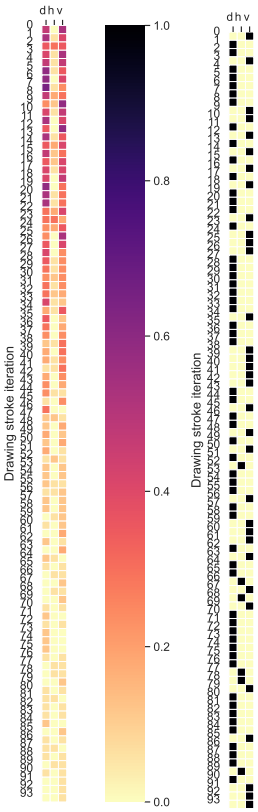

Figure 13. Hexagon illiterate sequences normalized and most popular

From both of the sequence heatmaps, it is apparent that the groups have made more strokes as a whole at the start of the sequence heatmap. Also, the horizontal segment is not as common in these sequences as it was in the multicube. The literate group has used diagonal lines more frequently than the illiterate group and the former has it as their most popular segment throughout the iterations.

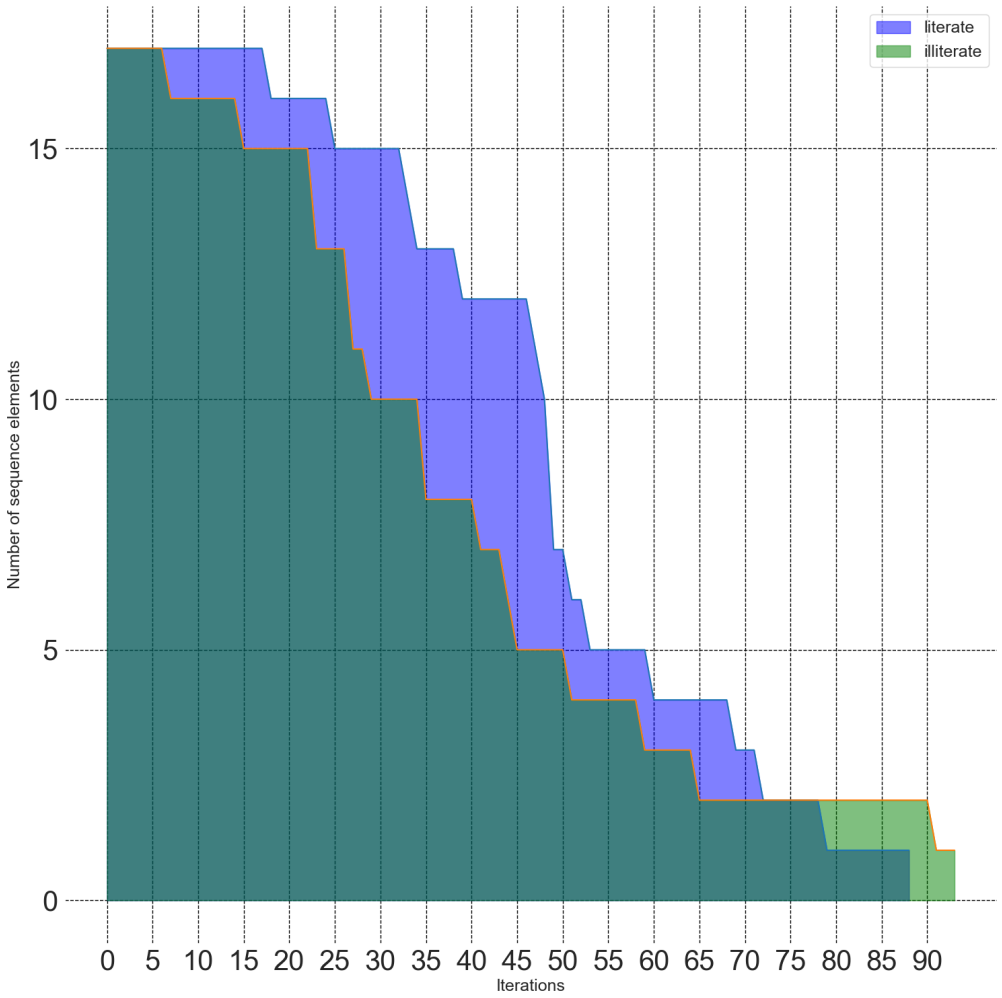

Figure 14. Hexagon Drawing Iterations Graph

The Figure 14 tells us that the literate did more drawing strokes in general while painting but the longest stroke count was in the illiterate group. Moreover, the literate group has a steep decline from 12 to 5 in participants drawing between the 45th and 5th iterations which means that 7 people finished their test in the rage of these strokes. On the other hand, the illiterate test group had a more steady decline which means that the amount of strokes taken varied more than in the literate group.

4.6.3 Patterns

For patterns an occurrence graph was constructed again but this time it has different patterns and frequencies compared to the multicube Figure 8.

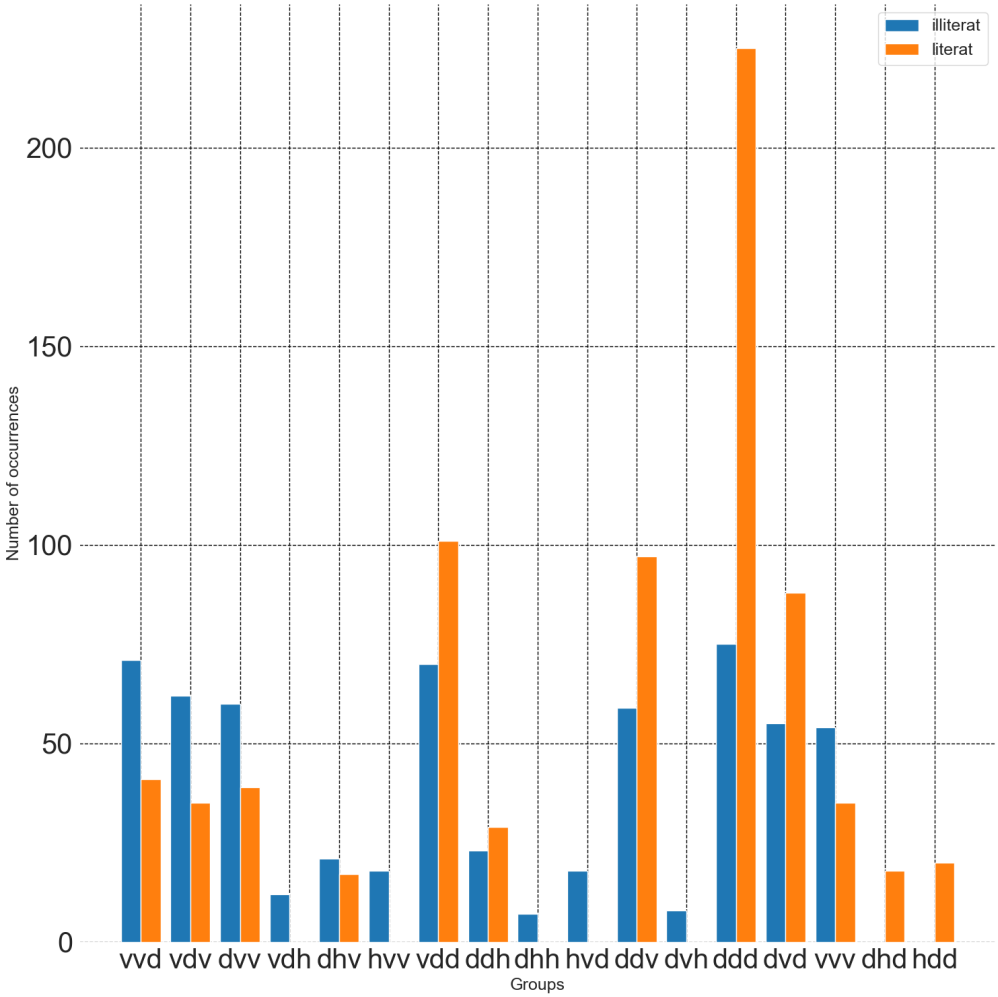

Figure 15. Hexagon Occurrences Graph

From the frequency graph, it can be seen that there are 10 overlapping patterns but their count is different between the two groups. Particularly the ddd element has appeared over 200 times in the literate group whereas the illiterate have used it less than 100 times. The patterns dhd and hdd have only appeared in the literate group and the vhd, hvv, dhh, hvd and dvh have been only used by the illiterate test subjects.

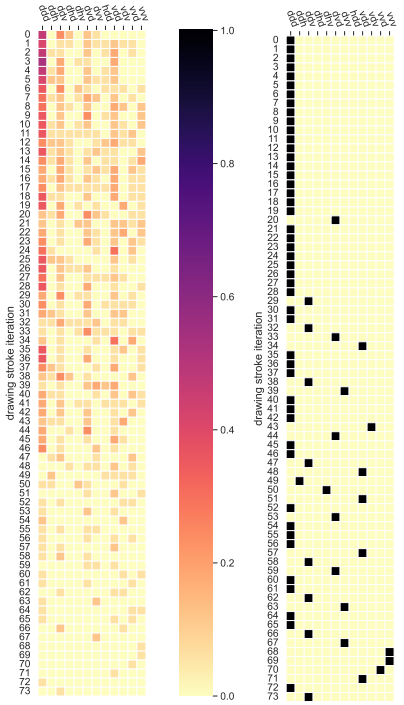

Figure 16. Hexagon literate patterns normalized and most popular

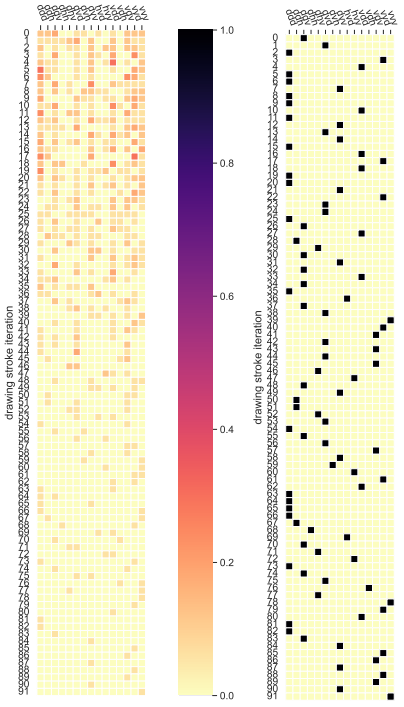

Figure 17. Hexagon illiterate patterns normalized and most popular

The pattern distribution gives a better view of how the used sequence patterns play out over time. It is apparent that the ddd pattern is the most prevalent in the literate heatmap, where in the first 30 iterations it is the most popular 28 times whereas, in the illiterate graph, it is only 10. It also reinforces the observations made in Figure 12 where ddd is the most used pattern overall.

From these results, it can be concluded that the methodology implemented can be used with other drawings that are done in the same JSON format.

4.7 Classification

To show that the results can be used in multiple ways, a classification algorithm is trained to demonstrate that if more data were to be gathered and we would have a type of person not determined beforehand then using the bigger dataset a prediction can be made which tells the class that the individual is in. The patterns were chosen as the features that are used in the training and prediction. For the classification Scikit-learn [38] Decision Tree [23] and Random Forest [22] are used.

4.7.1 Data

For training and testing purposes, a frequency table is made where each column is a pattern and each row is an individual. There are in total 14 columns and 36 rows, where 18 represent the literate and 18 are the illiterate test subjects. This table therefore represents how many times each individual uses a certain pattern.

4.7.2 Fisher score

Since there are in total of 14 features, some of which don't describe both of the subject groups, Fisher's score is used to determine the best descriptors to be used in the classification algorithm. This makes running the algorithm faster and overfitting less likely [24]. This was carried out in a way that the data was split into 3 equally distributed folds using StratifiedKFold so that the labels would have equal weight [46] from scikit-learn [38]. Each fold got a different set of 3 features from the Fisher's score method and then the features are evaluated using nested cross-validation [26] on a Decision Tree [23] to evaluate how good the features are. Some of the best pattern combinations using this method are shown in the table 2. The metrics here are obtained with the help of a classification report method provided by Scikit-learn.

Table 2. Feature testing results

Results			
Features	Precision	Recall	Accuracy
'hhh','hhd','vvd'	0.88	0.83	0.83
'vhh','hhh','hhd'	0.76	0.75	0.75
'vhh','vdh','dhv'	0.39	0.42	0.42

The best accuracy and precision were from the features hhh, hdd, vdd. It should be noted here that all of these patterns only occurred in the illiterate drawings and the other results - for instance, vhh, hhh, hdd - some patterns only happen in one group as well. In this case, it can be concluded that the best predictors are the ones that don't overlap between the two groups.

4.7.3 Random forest

To optimize for the best results for classification Random Forest algorithm is used. Random Forest is made up of multiple Decision Trees where the number of trees can be configured to optimize the results. For this optimization, Nested Cross-Validation is used with GridSearchCV and Cross-Validation Score from the Scikit-learn library. The number of trees tried for was from 1 to 101 with the step of 3. Table 3 shows the best result obtained.

Table 3. The best result achieved with Random Forest Nested Cross-Validation

Results				
Type	Precision	Recall	Accuracy	Number of Trees
Nested Cross-validation score	0.92	1.0	0.94	10

Graph 18 shows how the performance metrics are distributed with different number of trees in a Random Forest algorithm. The tree size range is the same as in the Random Forest subsection. The metrics are the same besides the three lower spikes that occurred before 22, at 58, and at 100. Since there is not a lot of data these results should not be taken too seriously, but still, it has been shown that with a larger dataset the training of a classification model is possible.

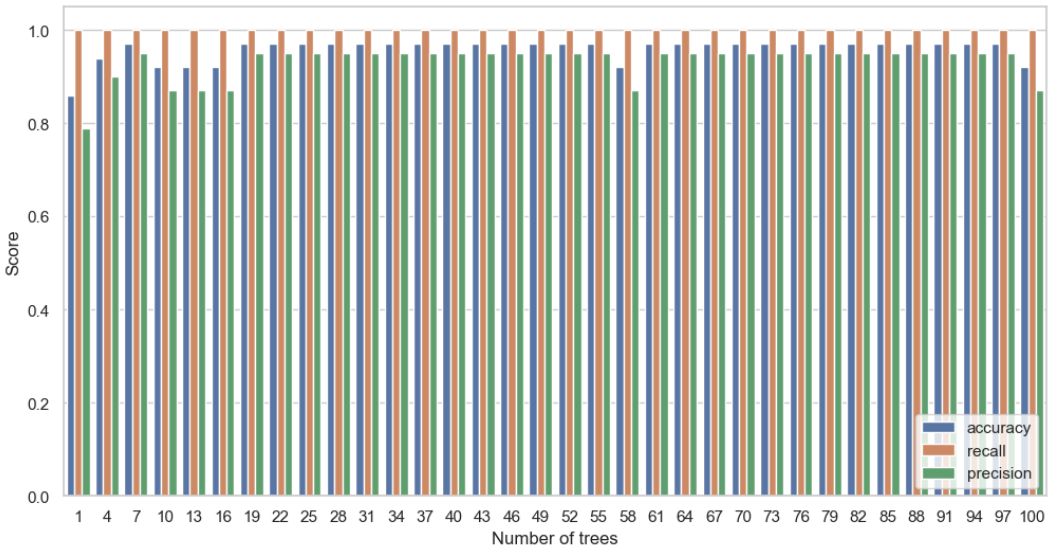

Figure 18. Nested cross-validation with different tree sizes

From this graph, it can be seen that by trying out different tree sizes the classification model is tunable and parameter tuning can be applied to get a better result.

5. Conclusions

This research aimed to find sequence patterns and strategies that groups with different educational backgrounds have on the multicube drawing test. To achieve this goal, a workflow with developed methodologies using a computer vision model and data mining technique was implemented.

In the first part, a way of describing the objects as an input to the computer vision model had to be constructed and a dataset corresponding to those requirements had to be made. Moreover, the previously mentioned set was used to train a YOLOv8 model that can detect and classify objects from pictures.

Furthermore, in the second part, a methodology had to be worked out where the object detection would classify the given lines in the pictures in the right order. This was done by splitting the pictures, based on pencil pressure, into smaller pictures that contain only one segment drawn as a single stroke. From these fragments, a sequence of elements was put together each of which describes how a drawing has been made in each iteration of a stroke. These sequences can be put together to visualize the group behavior in each drawing moment and to give a picture of how many strokes a group needed to finish the task.

In the next part, the Generalized Sequence Pattern algorithm was used to find common patterns in both of the study groups. There was a problem with this approach in that the patterns could not be validated and thus there is a problem that the results of this algorithm cannot be verified. After the patterns were found, an ordered sequence of patterns used was generated for each individual from the segment sequences found in part two. These patterns were formatted in a way that a pattern usage graph over the drawing iteration could be put together for the whole group. Moreover, the pattern frequency by group was constructed which describes the different popular elements used in the test drawing.

All of these methods were made so that they could be used in the future to be used on other drawing analysis-related work like cognitive disease or group behavior research. To prove this the author demonstrated the same methods work on different 2 group datasets where as a task a hexagon shape had been drawn.

The analysis concluded that by comparing both groups outlining differences but also a few similarities were found. The main notable features that set the two groups apart were the number of strokes made to finish the drawing, the patterns used by the group, and the pattern frequencies.

From the sequence heatmaps, it can be concluded that both the study groups took different approaches for drawing the tests where the literate had a greater variety in the distribution of initial drawing sequences and the illiterate used the diagonal stroke more than any other.

Figure 7 showed that while both the subject groups had a drop in the number of people drawing at one moment, the literate began to finish earlier than the illiterate where most of the former group had finished their drawing before the 15th stroke but the latter between and after the 15th and 20th strokes.

Moreover, in Figure 8 both of the study groups had a few unique patterns that they used. The literate had 4 and the illiterate 6 while there were also 4 overlapping patterns. The illiterate also have used some patterns way more than others where hhh, hvv and hhv have been used more than 15 times and where the other patterns are below that threshold. On the other hand, all of the literate pattern occurrences range between 5 and 10 which makes the distribution throughout the bar chart more equal.

The pattern heatmaps also showed a difference in approach where the frequency at which the patterns occur over time dropped in the literate group after 4th drawing iteration to 12.5%. This is in comparison to the illiterate heatmap, where there is a drop to 60% after the 6th iteration. Furthermore, the y-axis on both of the graphs differs between the two groups where the educated had 12 iterations at maximum and the less educated had 19.

In addition, it is evident from the Classification subsection in the Results that a machine-learning algorithm can be trained in the format presented in this thesis. There are also optimizations with the parameters where Fisher score determined that the best patterns that describe both of the groups are hhh, hhd, vvd and that the best number of trees required in a Random Forests model is 10.

In conclusion, the methodology and workflow were developed and the goal of this thesis was met. The segmentation and object detection were done in a way that the sequences would be generated in the correct order and in a format that could be processed by a text-based algorithm. Patterns were found with the help of the Generalized Sequence Pattern algorithm, even though it worked the results could not be validated. Lastly, the results were converted to a picture format which can be used by psychologists to more easily analyze the two groups. These methods and workflow were made so that they could be used on similar datasets to analyze other test groups in the future. The results achieved in this thesis show that the proposed methodology and workflow could be successfully implemented in this particular domain and be applied in the analysis of other drawing tests as well.

Acknowledgments

I would like to thank my supervisors professor Sven Nõmm for his encouragement, inspiration, and motivation, and professor Aaro Toomela for knowledge sharing and collaboration. Last but not least, I would like to express my deepest gratitude to my friends, family and especially my mother, for unconditional and unfailing support.

References

[1] Kurt Gottschaldt. “Über den Einfluß der Erfahrung auf die Wahrnehmung von Figuren: I. Über den Einfluß gehäufter Einprägung von Figuren auf ihre Sichtbarkeit in umfassenden Konfigurationen”. In: *Psychologische Forschung* 8 (1926), pp. 261–317.

[2] Hertha Kopfermann. “Psychologische Untersuchungen über die Wirkung zweidimensionaler Darstellungen körperlicher Gebilde.” In: *Psychologische forschung* (1930).

[3] Sven Nõmm et al. “Deep CNN Based Classification of the Archimedes Spiral Drawing Tests to Support Diagnostics of the Parkinson’s Disease”. In: *IFAC-PapersOnLine* 53.5 (2020). 3rd IFAC Workshop on Cyber-Physical & Human Systems CPHS 2020, pp. 260–264. ISSN: 2405-8963. DOI: https://doi.org/10.1016/j.ifacol.2021.04.185. URL: https://www.sciencedirect.com/science/article/pii/S2405896321003360.

[4] Sven Nõmm et al. “Detailed Analysis of the Luria’s Alternating SeriesTests for Parkinson’s Disease Diagnostics”. In: *2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA)*. 2018, pp. 1347–1352. DOI: 10.1109/ICMLA.2018.00219.

[5] C. Marquardt and N. Mai. “A computational procedure for movement analysis in handwriting”. In: *Journal of Neuroscience Methods* 52.1 (1994), pp. 39–45. ISSN: 0165-0270. DOI: http://dx.doi.org/10.1016/0165-0270(94)90053-1.

[6] Peter Drotar et al. “Evaluation of handwriting kinematics and pressure for differential diagnosis of Parkinson’s disease”. In: *Artificial Intelligence in Medicine* 67 (2016), pp. 39–46. ISSN: 0933-3657. DOI: https://doi.org/10.1016/j.artmed.2016.01.004.

[7] Elli Valla et al. “Tremor-related feature engineering for machine learning based Parkinson’s disease diagnostics”. In: *Biomedical Signal Processing and Control* 75 (2022), p. 103551.

[8] Clayton R Pereira et al. “Deep learning-aided Parkinson’s disease diagnosis from handwritten dynamics”. In: *2016 29th SIBGRAPI conference on graphics, patterns and images (SIBGRAPI)*. IEEE. 2016, pp. 340–346.

[9] Zoltan Galaz et al. “Comparison of CNN-Learned vs. Handcrafted Features for Detection of Parkinson’s Disease Dysgraphia in a Multilingual Dataset”. In: *Frontiers in Neuroinformatics* 16 (2022). ISSN: 1662-5196. DOI: 10.3389/fninf.2022.877139. URL: https://www.frontiersin.org/articles/10.3389/fninf.2022.877139.

[10] Xuechao Wang et al. “Comparison of one- two- and three-dimensional CNN models for drawing-test-based diagnostics of the Parkinson’s disease”. In: *Biomedical Signal Processing and Control* 87 (2024), p. 105436. ISSN: 1746-8094. DOI: https://doi.org/10.1016/j.bspc.2023.105436. URL: https://www.sciencedirect.com/science/article/pii/S1746809423008698.

[11] Elli Valla et al. “Deep Learning Based Segmentation of Luria’s Alternating Series Test to Support Diagnostics of Parkinson’s Disease”. In: *Proceedings of the 22nd International Conference on Machine Learning and Application, IEEE, ICMLA2023*. 2023, Accepted.

[12] Glenn Jocher, Ayush Chaurasia, and Jing Qiu. *Ultralytics YOLOv8*. Version 8.0.0. 2023. URL: https://github.com/ultralytics/ultralytics.

[13] Abhinav Rai. “An Overview of Association Rule Mining & its Applications”. In: (2022). URL: https://www.upgrad.com/blog/association-rule-mining-an-overview-and-its-applications/.

[14] Jekaterina Viltšenko. “Kella joonistamise testi semantiliste ja kinemaatiliste parameetrite analüüs”. Tallinn University of Technology, 2021.

[15] Henry Laur. *Automated Segmentation and Semantic Analysis of Writing and Drawing Tests for Parkinson’s Disease Diagnostics*. 2021.

[16] Carol Huntsinger et al. “Cultural differences in Chinese American and European American children’s drawing skills over time”. In: *Early Childhood Research Quarterly* 26 (Jan. 2010), pp. 240–265. DOI: 10.1016/j.ecresq.2010.04.002.

[17] Sam M.S. N. *PsychologyDictionary.org*. URL: https://psychologydictionary.org/strategy/.

[18] Ilja Mašarov. “Digital Clock Drawing Test Implementation and Analysis”. Master thesis. Tallinn University of Technology, 2017.

[19] Berrin Yanikoglu, Aytac Gogus, and Emre Inal. “Use of handwriting recognition technologies in tablet-based learning modules for first grade education”. In: *Educational Technology Research and Development* 65 (July 2017). DOI: 10.1007/s11423-017-9532-3.

[20] Published by Tableau. “How Data Mining Works: A Guide”. In: (2023). URL: www.tableau.com/learn/articles/what-is-data-mining.

[21] Domingos Pedro. “A Few Useful Things to Know About Machine Learning”. In: *Communications of the ACM* 55 (Oct. 2012). DOI: 10.1145/2347736.2347755.

[22] “Random Forests”. In: *Machine Learning 45* (2001), pp. 5–32. DOI: https://doi.org/10.1023/A:1010933404324.

[23] Geeks for Geeks. “Decision Tree”. In: (2023). URL: https://www.geeksforgeeks.org/decision-tree/.

[24] Quanquan Gu, Zhenhui Li, and Jiawei Han. “Generalized Fisher Score for Feature Selection”. In: *CoRR* abs/1202.3725 (2012). arXiv: 1202.3725. URL: http://arxiv.org/abs/1202.3725.

[25] Rahul Shah. “Tune Hyperparameters with GridSearchCV”. In: (2023). URL: https://www.analyticsvidhya.com/blog/2021/06/tune-hyperparameters-with-gridsearchcv/.

[26] Jason Brownlee. “Nested Cross-Validation for Machine Learning with Python”. In: (2021). URL: https://machinelearningmastery.com/nested-cross-validation-for-machine-learning-with-python/.

[27] Guido Van Rossum and Fred L Drake Jr. *Python reference manual*. Centrum voor Wiskunde en Informatica Amsterdam, 1995.

[28] *Anaconda Software Distribution*. Version Vers. 2-2.4.0. 2020. URL: https://docs.anaconda.com/.

[29] Thomas Kluyver et al. “Jupyter Notebooks – a publishing format for reproducible computational workflows”. In: *Positioning and Power in Academic Publishing: Players, Agents and Agendas*. Ed. by F. Loizides and B. Schmidt. IOS Press. 2016, pp. 87–90.

[30] J. D. Hunter. “Matplotlib: A 2D graphics environment”. In: *Computing in Science & Engineering* 9.3 (2007), pp. 90–95. DOI: 10.1109/MCSE.2007.55.

[31] Charles R. Harris et al. “Array programming with NumPy”. In: *Nature* 585.7825 (2020), pp. 357–362. DOI: 10.1038/s41586-020-2649-2. URL: https://doi.org/10.1038/s41586-020-2649-2.

[32] Wes McKinney. “Data Structures for Statistical Computing in Python”. In: *Proceedings of the 9th Python in Science Conference*. Ed. by Stéfan van der Walt and Jarrod Millman. 2010, pp. 56–61. DOI: 10.25080/Majora-92bf1922-00a.

[33] Adam Paszke et al. “PyTorch: An Imperative Style, High-Performance Deep Learning Library”. In: *Advances in Neural Information Processing Systems 32*. Curran Associates, Inc., 2019, pp. 8024–8035. URL: http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf.

[34] Michael L. Waskom. “Seaborn: statistical data visualization”. In: *Journal of Open Source Software* 6.60 (2021), p. 3021. DOI: 10.21105/joss.03021. URL: https://doi.org/10.21105/joss.03021.

[35] Jackson Antonio do Prado Lima. *GSP-Py - Generalized Sequence Pattern algorithm in Python*. May 2020. DOI: 10.5281/zenodo.3333987. URL: https://doi.org/10.5281/zenodo.3333987.

[36] Satheeshkatipomu. *Nms-python*. 2019. URL: https://github.com/satheeshkatipomu/nms-python/tree/master.

[37] Boris Sekachev et al. *Opencv/cvat: v1.1.0*. Version v1.1.0. Aug. 2020. DOI: 10.5281/zenodo.4009388. URL: https://doi.org/10.5281/zenodo.4009388.

[38] F. Pedregosa et al. “Scikit-learn: Machine Learning in Python”. In: *Journal of Machine Learning Research* 12 (2011), pp. 2825–2830.

[39] Kinza Yasar. “PyTorch”. In: (2022). URL: https://www.techtarget.com/searchenterpriseai/definition/PyTorch.

[40] GeeksforGeeks. “Introduction to Convolution Neural Network”. In: (2023). URL: https://www.geeksforgeeks.org/introduction-convolution-neural-network/.

[41] Pintusaini(username). “Generalized Sequential Pattern (GSP) Mining in Data Mining”. In: (2023). URL: https://www.geeksforgeeks.org/generalized-sequential-pattern-gsp-mining-in-data-mining/.

[42] Chinmay Bhalerao. “A Deep Dive into Non-Maximum Suppression[NMS]: Understanding the Math Behind Object Detection”. In: (2023). URL: https://medium.com/@BH_Chinmay/a-deep-dive-into-non-maximum-suppression-nms-understanding-the-math-behind-object-detection-765ff48392e5.

[43] Manasa Ramakrishnan. “What is Classification in Machine Learning and Why is it Important?” In: (2022). URL: https://emeritus.org/blog/artificial-intelligence-and-machine-learning-classification-in-machine-learning/.

[44] Nathan Rosidi. “Feature Selection Techniques in Machine Learning”. In: (2023). URL: https://medium.com/mlearning-ai/feature-selection-techniques-in-machine-learning-82c2123bd548.

[45] Jason Brownlee. “A Gentle Introduction to k-fold Cross-Validation”. In: (2023). URL: https://machinelearningmastery.com/k-fold-cross-validation/#:~:text=Cross%2Dvalidation%20is%20a%20statistical,skill%20of%20machine%20learning%20models..

[46] Geeks for Geeks. “Stratified K Fold Cross Validation”. In: (2023). URL: https://www.geeksforgeeks.org/stratified-k-fold-cross-validation/.

[47] Jaime Arboleda Castilla. “A step by step guide to Nested Cross-Validation”. In: (2021). URL: https://www.analyticsvidhya.com/blog/2021/03/a-step-by-step-guide-to-nested-cross-validation/.

Appendix 1 – Non-Exclusive License for Reproduction and Publication of a Graduation Thesis[1]

I Peeter Tarvas

1. Grant Tallinn University of Technology free licence (non-exclusive licence) for my thesis "Drawing Strategies Analysis for Embedded Figure Drawing Tests", supervised by Sven Nõmm and Aaro Toomela
   1.1. to be reproduced for the purposes of preservation and electronic publication of the graduation thesis, incl. to be entered in the digital collection of the library of Tallinn University of Technology until expiry of the term of copyright;
   1.2. to be published via the web of Tallinn University of Technology, incl. to be entered in the digital collection of the library of Tallinn University of Technology until expiry of the term of copyright.
2. I am aware that the author also retains the rights specified in clause 1 of the non-exclusive licence.
3. I confirm that granting the non-exclusive licence does not infringe other persons' intellectual property rights, the rights arising from the Personal Data Protection Act or rights arising from other legislation.

16.01.2024

[1]The non-exclusive licence is not valid during the validity of access restriction indicated in the student's application for restriction on access to the graduation thesis that has been signed by the school's dean, except in case of the university's right to reproduce the thesis for preservation purposes only. If a graduation thesis is based on the joint creative activity of two or more persons and the co-author(s) has/have not granted, by the set deadline, the student defending his/her graduation thesis consent to reproduce and publish the graduation thesis in compliance with clauses 1.1 and 1.2 of the non-exclusive licence, the non-exclusive license shall not be valid for the period.