SKETCH RECOGNITION BASED CLASSIFICATION FOR EYE MOVEMENT BIOMETRICS IN VIRTUAL REALITY

A Thesis

by

SIDDHARTH SUBRAMANIYAM

Submitted to the Office of Graduate and Professional Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

Chair of Committee,	Tracy Hammond
Committee Members,	Joanna Lahey
	John Keyser
Head of Department,	Dilma Da Silva

August 2019

Major Subject: Computer Science

ABSTRACT

Virtual Reality (VR) is a computer technology that uses computer-generated simulations of a three-dimensional environment that can be interacted with special electronic equipment such as a head-mounted display (HMD). VR users have grown from 5 million in 2015 to over 170 million in 2018. However, authentication systems for virtual reality is relatively underexplored. We propose an authentication method for VR that augments the traditional password-based system with biometric security.

Biometrics is an active area of research in the HCI, pattern recognition, and machine learning communities. Various physiological features such as fingerprint, DNA, iris pattern, and facial recognition are widely used in biometrics. Recently, there has been an interest in using eye movement patterns, gait, and keystroke signature as behavioral biometric modalities.

As most VR systems use a head-mounted display, it is difficult to use standard biometric authentication systems such as fingerprint, face recognition, etc. Eyes are a natural mode of interaction in the VR domain. Furthermore, inaccuracies due to head movement are eliminated in the VR domain due to head-mounted VR headsets. In this work, we use the human eye as a source of biometric information for VR and augment a password-based authentication system with it. Latent features encoded by behavioral characteristics of individuals are present in eye movement data. This information combined with other authentication methods can create effective security systems for Virtual Reality. We use a head-mounted display with a built-in eye tracker to collect eye movement information. We analyzed the data by extracting eye movement and sketch recognition based features to build a classification system. We obtained results with accuracy values of up to 50%.

CONTRIBUTORS AND FUNDING SOURCES

Contributors

This work was supported by a thesis committee consisting of Dr. Tracy Hammond (advisor) and Dr. John Keyser of the Department of Computer Science & Engineering and Dr. Joanna Lahey of the Department of Public Service and Administration.

All work for this thesis was completed by the student in collaboration with Chinmay Phulse, an Undergraduate honors thesis student in the Department of Computer Science. I was responsible for designing the features and algorithms for the statistical analysis. Chinmay built the authentication and data collection interface. We both conducted user studies together and performed the analysis mentioned in the results section.

All other work conducted for the thesis was completed by the student independently.

TABLE OF CONTENTS

Page

LIST OF FIGURES

LIST OF TABLES

1. INTRODUCTION

1.1 Virtual Reality and Authentication (VR)

Virtual Reality (VR) is a computer technology that uses computer-generated simulation of a three-dimensional image or environment that can be interacted with in a seemingly real or physical way by a person using special electronic equipment, such as a head-mounted display (HMD). VR technology has numerous applications in education, training, entertainment, military, security, and more [15]. The use of HMDs in the entertainment and gaming sectors has been driving the growth of the VR market. VR systems have grown from 5 million users in 2015 to 170 million in 2018 [16]. As computers become faster, and real-world simulations get better, VR is bound to become a mainstay in homes and offices [17].

While VR has grown exponentially, usable security for VR systems has been relatively underexplored. There is little literature that explores authentication mechanisms in VR. This is mainly due to the limited interaction model in VR. As VR systems are head-mounted, using physical devices for authentication is sub-optimal.

As the use of biometrics for authentication is becoming common, it is worth exploring the feasibility and effectiveness of using biometric methods for security in the VR domain.

1.2 Introduction to Biometrics and Human Identification [1]

Biometrics refers to metrics related to human characteristics. It is used for *Human Identifications* and is an active area of research in the pattern recognition and machine learning community. Human identification is done in an extensive number of ways for various applications such as forensics, law enforcement, surveillance, access control, etc.

[1] Part of this section is reprinted with permission from [18] "A score level fusion method for eye movement biometrics" by A. George and A. Routray, Pattern Recognition Letters, vol. 82, pp. 207 - 215, 2016.

Figure 1.1: VR headset. Reprinted from [1]

[18]. Physiological features such as DNA, iris pattern and facial recognition [19] are widely used in biometrics. Though some biometric modalities like brain signals and heartbeats have been proposed, their invasive nature limits their applications.

Recently, several behavioral biometric modalities have been proposed such as eye movement, gait, keystroke dynamics, and typing patterns [20]. They are also referred to as *weak biometrics*, as they remain unique only over small groups. But their connection to conscious and emotional state makes them difficult to replicate [21]. These biometrics have previously been combined to create or augment existing authentication systems [22].

An effective biometric system should have features following these guidelines [19, 18]:

1. The features should be unique for each individual
2. They should not change with time (template aging effects)
3. Acquisition of parameters should be easy (low computational complexity and non-

invasive)

4. Accurate and automated algorithms should be available for classification

5. Counterfeit resistance

6. Low cost

7. Ease of implementation.

Eye movement is one of the most common human actions. Every second, a human eye makes around 3 *saccadic movements* - a rapid movement of the eye between fixation points [23]. Biometrics using patterns obtained from eye movements is a relatively new field of research. Compared to other biometric systems that rely on physiological characteristics of the human body, eye movement based biometrics tries to identify the behavioral patterns as well as information regarding physiological properties of the tissues and muscles generating eye movements [24]. Due to the composition of human eye muscles, there are reliable individual differences in the characteristics of oculomotor measures: saccades and smooth-pursuit eye movements [25, 26, 27, 28, 29, 30, 31]. It has also been suggested that these oculomotor measures (measures obtained from eye movement) can be used for biometric identification systems. [25, 32, 33].

1.3 Eye Movement Biometrics for Virtual Reality

As VR headsets are head mounted, the use of common interaction methods such as keyboard and mouse are limited. It is also hard to use the physiological characteristics of the human body as viable biometric methods. Face recognition is not feasible due to the presence of a headset. Eyes are a natural way to interact in VR. This makes it worthwhile to explore the use of eye movement for authentication in virtual reality.

Eye tracking devices can be used to measure these eye movement patterns. These devices project infrared light onto the eyes and capture the reflection of the light (infrared oculography [2], see figure 1.2) [34, 35]. Eye trackers, in general, are sensitive to head movement and require recalibration. Small movements of the head can make the resulting measurements inaccurate. Also, the user is required to recalibrate every time they perform a task. In the VR domain, these inaccuracies due to head/body position are eliminated.

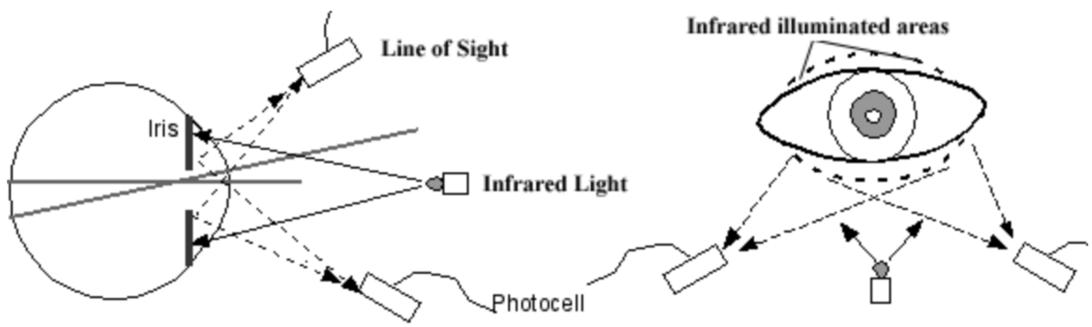

Figure 1.2: Principle of infrared-oculography. Reprinted from [2]

In our study, we would like to measure and verify these oculomotor measures using the FOVE0 VR Headset (Figure 1.3). The FOVE0 has an eye tracker inside the VR headset.

1.4 Sketch Recognition and Eye Movement

Sketch Recognition is the automated recognition of hand-drawn diagrams [36]. It is at the crossroads of AI and Human-Computer Interaction. Many algorithms have been developed to identify sketches [37, 38], both general freehand sketches and domain-specific sketches [39, 40, 41].

A sketch is made up of *strokes* [36]. A stroke is defined as the path from pen down to pen up on the screen. It is a series of **x, y, time** values. The first point is represented by (x_0, y_0, t_0) where *x* and *y* represent spatial coordinates of the point in 2 dimensions

Figure 1.3: FOVE headset with inbuilt eye tracker. Reprinted from [3]

and *t* represents a time value. The last point is given by $(x_{n-1}, y_{n-1}, t_{n-1})$ where *n* is the number of points in the stroke. (See Figure 1.4) Various algorithms (described below) are then used on these strokes to develop features and perform recognition tasks.

Sketch recognition algorithms generally fall into three categories:

- **Appearance-based methods** These refer to algorithms from the computer vision field where the entire sketch is treated as an image.

- **Geometry-based methods** These algorithms breakdown sketches into geometric

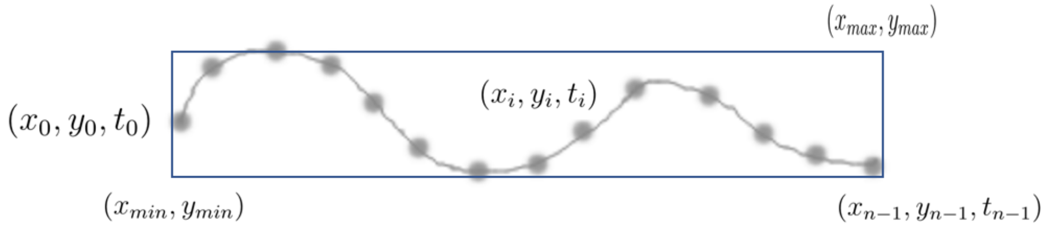

Figure 1.4: Example stroke. x, y and time values were sampled as the stroke was drawn.

shapes and then identifies these shapes. [42, 43]

- **Gesture-based methods** These methods identify the path of the pen to perform recognition. [44, 45]

This idea of stroke is not specific to pen and paper, it can be used for analyzing eye movement data as well. Eye tracking devices provide gaze data in the form of a stroke, consisting of a set of (x_i, y_i, t_i) coordinates. This allows us to apply algorithms from the sketch recognition field for eye tracking.

In most studies, eye movement data are analyzed using purpose-built saccade algorithms [25, 18]. In our study, we treat the measures as a sketch, which allows us to use any of the standard sketch recognition techniques available. This will also help us develop features that can be fed into various machine learning techniques.

2. PRIOR AND RELATED WORK

2.1 Virtual Reality Authentication

The area of security for virtual reality is relatively underexplored. This lack can mainly be attributed to the limitation of using common interaction methods such as keyboard and mouse. As VR systems are head-mounted, user interaction with physical devices become difficult. This limits the use of biometric devices such as fingerprint reader, RFID, etc.

A common authentication method used in VR is by interacting with objects in the simulated environment [46, 47]. Three password-based authentication mechanisms for VR were compared by [4] using PIN and Pattern lock, redesigned for the VR environment. Figure 2.1 shows the three proposed systems: 3D patterns, 2D sliding patterns, and a PIN system. Though 3D patterns were found to have the highest security level, pattern lock and PIN systems were perceived to be more usable. As these are not biometric methods, they can be spoofed and easily replicated by malicious users.

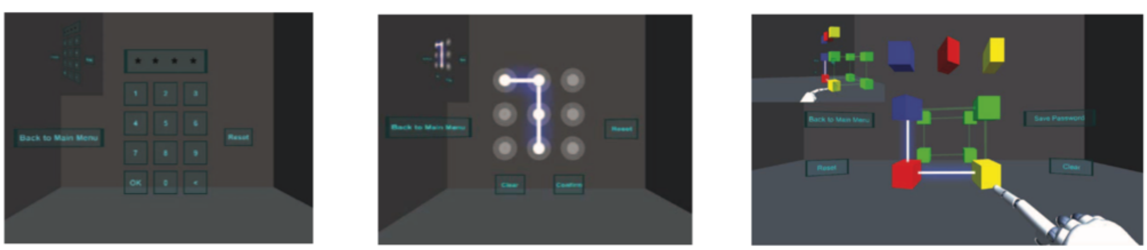

Figure 2.1: PIN, 2D and 3D password system in VR. Reprinted from [4] ©IEEE 2016

A proposed biometric method for VR uses gait for authentication [48]. While this method is effective, it requires the user to walk around which may not work well in small spaces where movement is limited or not feasible. In this work, we address these issues by combining the familiar password based methods with eye movement biometrics.

2.2 Biometrics and Eye Tracking[1]

Biometric identification and verification systems have been shown to be more secure than a traditional password based security system because passwords can be easily shared but biometric features are not as easy to share or duplicate [26]. The application of a biometric authentication technique using keystroke identification was discussed by [28]. This has been determined to be a useful feature, especially when combined with other layers of authentication.

Eye movements have been utilized as a biometric security feature with varying levels of results. Eye movement can be captured by using a camera and an infrared light, making it a low-cost option for the inclusion of biometric features[49]. Previous work has focused on gaze input for passwords because of their ease of use for both the user and security system as well as their resistance to attacks [50].

Generally, eye tracking studies can be categorized into three wide classes based on the stimulus used:

- **Image based**: Subjects are asked to look at images, while their eye movement data is recorded.
- **Text based**: In this case, subjects read sentences and their eye movement is data is recorded.
- **Video based**: Subjects are asked to perform a particular task that relates to seeing a video on the screen.

[1]Part of this section is reprinted with permission from [18] "A score level fusion method for eye movement biometrics" by A. George and A. Routray, Pattern Recognition Letters, vol. 82, pp. 207 - 215, 2016.

The type of study determines which eye movement features were used, as these features depend on the type of task being performed.

Early work by Kasprowski and Ober [51] was done to use eye movements as a biometric modality. Users were asked to follow a jumping dot on the screen, in a 3 by 3 matrix. These dots were shown at fixed time intervals, set at 100 - 200 ms. Features such as reaction time, stablization time were computed. Furthermore, frequency domain features were computed using signals from the data. Different classification methods such as naive Bayes, support vector machines (SVMs), KC45 decision trees, and KNNs were used and accuracy values of upto 82% were achieved. These results motivated further research in eye movement biometrics.

Bednarik et al. [52] conducted experiments on several tasks including text reading, moving cross stimulus tracking, and free viewing of images. They used FFT and PCA on the eye movement data. Several combinations of such features were tried. However, the best results were obtained using the distance between eyes which is not related to eye dynamics. Komogortsev et al. [33] used an Oculomotor Plant Mathematical Model (OPMM) to model the complex dynamics of the oculomotor plant. The plant parameters were identified from the eye movement data. This approach was further extended in [33]. Holland and Komogortsev [33] evaluated the applicability of eye movement biometrics with different spatial and temporal accuracies and various types of stimuli. Several parameters of eye movements were extracted from fixations and saccades. Weighted components were used to compare different samples for biometric identification. A temporal resolution of 250 Hz and a spatial accuracy of 0.5 degrees were identified as the minimum requirements for accurate gaze-based biometric systems. Kinnunen et al. [53] presented a task-independent user authentication system based on eye movements. Gaussian mixture modeling of short-term gaze data was used in their approach. Even though the accuracy rates were fairly low, the study opened up possibilities for the development of task-independent eye movement-based verification systems. Rigas et al. [54] explored variations in individual gaze patterns while observing human face images. Eye movements resulted were analyzed using a graph-based approach. The Multivariate Wald-Wolfowitz runs test was used to classify the eye movement data. This method achieved 70% rank-1 IR and 30% EER on a database of 15 subjects. Rigas et al. [55] extended this method using features of velocity and acceleration calculated from fixations. The feature distributions were compared using the Wald-Wolfowitz test.

Some studies have combined data from multiple stimuli for their studies. In [56], users were asked to perform tasks based on images or videos shown to them. This incentivizes the user to pay attention to the video and understand what was shown. Their study used features from both fixation/saccade data and clustered X-Y coordinates. Similarly, [22] combined eye tracking and keystroke data for biometrics.

Eye movement-based user authentication has also been studied in a desktop environment to counter should-surfing attacks [57, 58, 59]. A Gaze analysis technique for user identification was proposed by [60] that used human faces as stimuli and extracted features based on regions of interest in the image.

In a lot of studies, subjects get accustomed to the stimulus text or image, which leads to lower ocular activity. This problem is known as the *learning effect*. In order to circumvent this problem, a video is used as a stimulus. Such a study by [61] using features such as acceleration, geometric and muscular properties of the eye, obtained over 80% accuracy values. This also shows that eye movement input can be combined with other properties to improve overall system performance.

All of the above-mentioned studies use eye based features (such as saccade and fixation features). While these methods obtain good results, it is worth exploring the sketch recognition domain for features that treat eye movement patterns as strokes.

2.3 Eye Tracking and Sketch Recognition

Sketch recognition uses machine learning techniques to recognize the shape of hand-drawn sketches. As mentioned before the recognition techniques rely on geometry-based algorithms, appearance-based algorithms, and gesture-based algorithms to find and extract certain unique features of the shape [62, 43].

Alamudun et al. [10] used sketch recognition and eye tracking to predict the user identity and level of expertise as radiologists viewed x-ray images. The study inferred expertise of users studying mammograms. Subjects wore a head-mounted device that recorded their eye movements as well as other eye features such as pupil size and distance between pupils. Six sketch based features and seven eye based features were derived from the eye tracking data, and the F-score (the harmonic mean of precision and recall) for sketch-based recognition was 0.89 on average while the F-score based on eye tracking data was 0.7. This study used geometric and gesture-based features. The sketch-based features included the length and angle of gesture, duration, and spatial efficiency while the eye features included pupil size, scan path length, and the number of fixations. This resulted in an average of 6.1% false acceptance rate and a false rejection rate of 5%. Alamudun was the first to combine eye tracking and sketch recognition techniques for user identification [63, 64, 65, 66].

In this work, we augment the eye movement feature space by treating eye movement data as sketches and explore the use of sketch-based features to capture new information that helps with user identification.

3. EYE TRACKING

3.1 Introduction to Eye Tracking

Eye tracking is the process of measuring one's eye gaze, also known as the point of gaze. Eye trackers are used to carrying out these measurements and they record the position of the eyes and the movements they make [67, 5].

Eye movement measurement is typically done based on two methods [68]:

- Electro-oculography
- Infrared-oculography

Electro-oculography uses electrodes placed on different sides of the eye and measures change in voltage [69]. This method is both intrusive and not very accurate. Most modern-day eye trackers rely on infrared-oculography.

In infrared-oculography, infrared light is projected towards the pupil, causing reflection in both the cornea (outer-most optical element of the eye) and the pupil. The vector between them is tracked by an infrared camera. This method of optical tracking of reflections, is known as *pupil center corneal reflection (PCCR)* [70]. The gaze direction is determined by the relative difference in location of the pupil and corneal reflections (See Figure 3.1, 3.2).

The light source is in the infrared spectrum as the accuracy of the measurement depends on the identification of the pupil and corneal reflections. Visible spectrum light sources (with ordinary cameras) cannot provide the required contrast, affecting measurement accuracy. Furthermore, infrared light does not cause specular reflection (unlike visible light) allowing for easier differentiation between pupil and iris. As infrared is not in the visible spectrum, it does not distract the user while eyes are being tracked.

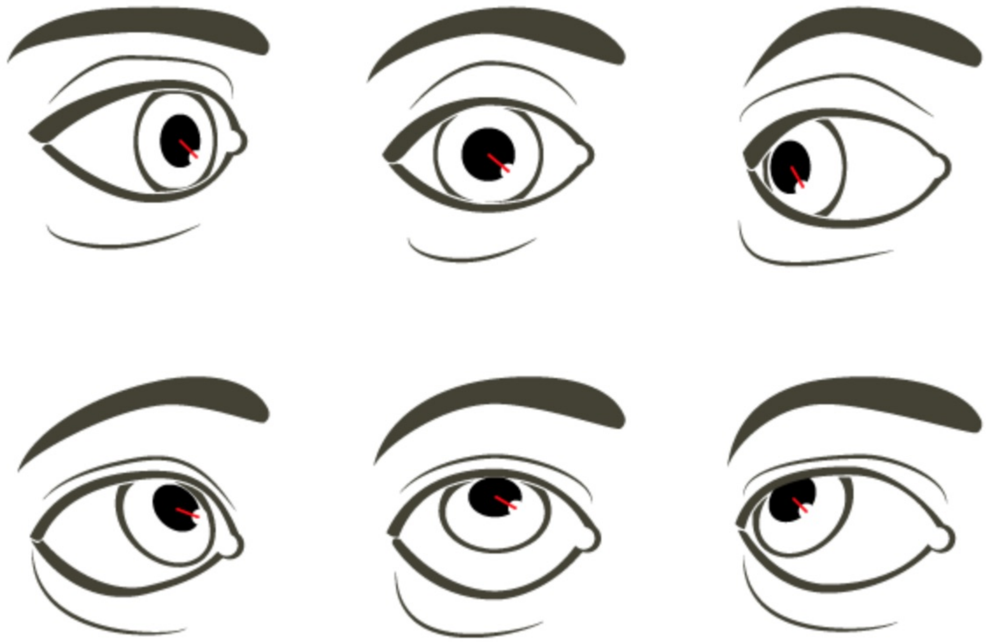

Figure 3.1: For different gaze directions, the vector between pupil and corneal reflection changes. Reprinted with permission from [5]

3.2 Eye Tracking System Requirements

An eye tracking system that captures and analyzes eye movement data requires the following:

- **Gaze Stimuli** This is required in order to cause significant eye movement that can be measured. These include text, stationary/moving objects, videos, etc. The choice of stimuli depends on the requirements of the study or system.
- **Eye Tracking device** These devices perform PCCR (explained above) and track the subject's eye. Typically, these devices provide data in the form of X-Y iris coordinates, time, eye velocity and acceleration. These devices can be Remote or Head-

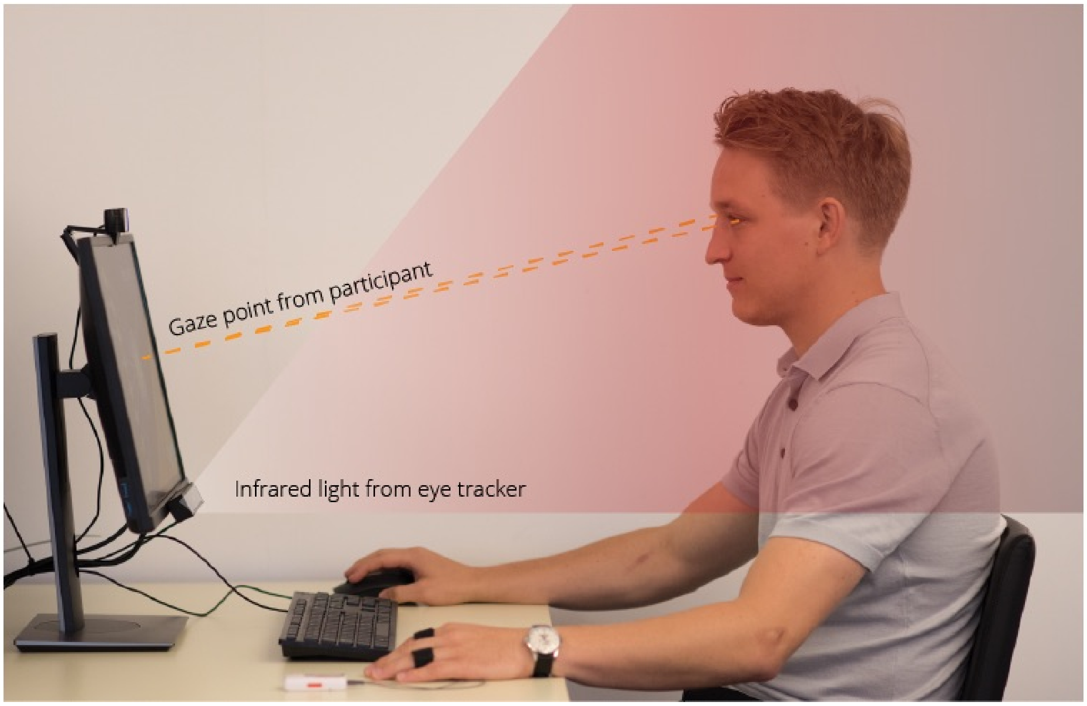

Figure 3.2: Remote eye tracker. Reprinted with permission from [5]

mounted. Head-mounted eye trackers must be worn, which makes them intrusive but are typically more accurate. Remote eye trackers are mounted on the desk and track the user.

- **Eye Data Features** As the data obtained from eye trackers are in the form of XY coordinates, they need to be processed into useful features so that can be used for further analysis.

3.3 Eye Movement

Physiological features of the human eyes provide a rich source of information about an individual. They are commonly used in user identification systems using traits from iris and retina scans. However, the eyes also contain behavioral characteristics in its movements, which are often ignored.

Biometrics from eye movement pattern is a relatively new area of research. These biometrics try to identify the behavioral patterns of an individual. Among all human actions, human eye movement is the most common. Every second we make approximately three of the rapid, stereotyped movements known as *saccades* [25]. Saccades are the path taken by the eye to get from one dwell spot to the next (known as fixation). These paths are not straight. This is because the muscles in the eyes travel horizontally and vertically only. The strength of these muscles is impacted by each individuals activity over time, making them unique.

While physiological traits do not vary greatly due to behavioral factors, eye movement depends on the psychological state, the stimulus for the movement and type of task being performed [71]. This makes it hard to replicate. Visual attention has been closely related to eye movement [72]. Thus, eye movement data could provide us with useful information to identify individuals.

Many studies have shown the connection between ocular movement and individual characteristics. For example, artists can get feedback on their work by finding regions of interest and other relevant data [73]. Using pseudo-random images as stimuli, eye movement is linked with attention and motivation [74]. Eye movement has also been linked to personality and the "Big Five" traits [75] (Neuroticism, Extraversion, Intellect, Agreeableness, Conscientiousness). Eye movement features based on fixations and dwell time were related to these traits.

3.4 Eye Trackers

Broadly, eye trackers can be divided into 2 types:

- Head-mounted eye trackers (Figure 3.3)
- Remote eye trackers (Figure 3.4)

Figure 3.3: Head-mounted eye tracker. Reprinted with permission from [5]

Head-mounted eye trackers are mobile devices fitted near the eyes and allow participants to move freely. This is typically more accurate in a natural environment with a lot of head movement. Screen-based eye trackers require the subject to sit in front of a monitor and interact with screen-based content. These track the eyes within certain limits and the freedom of movement is relatively small compared to head-mounted devices.

3.5 Eye Movement Features

Eye movement features are derived from the basic eye movement data obtained from the eye tracker. They are generally divided into two categories [76]:

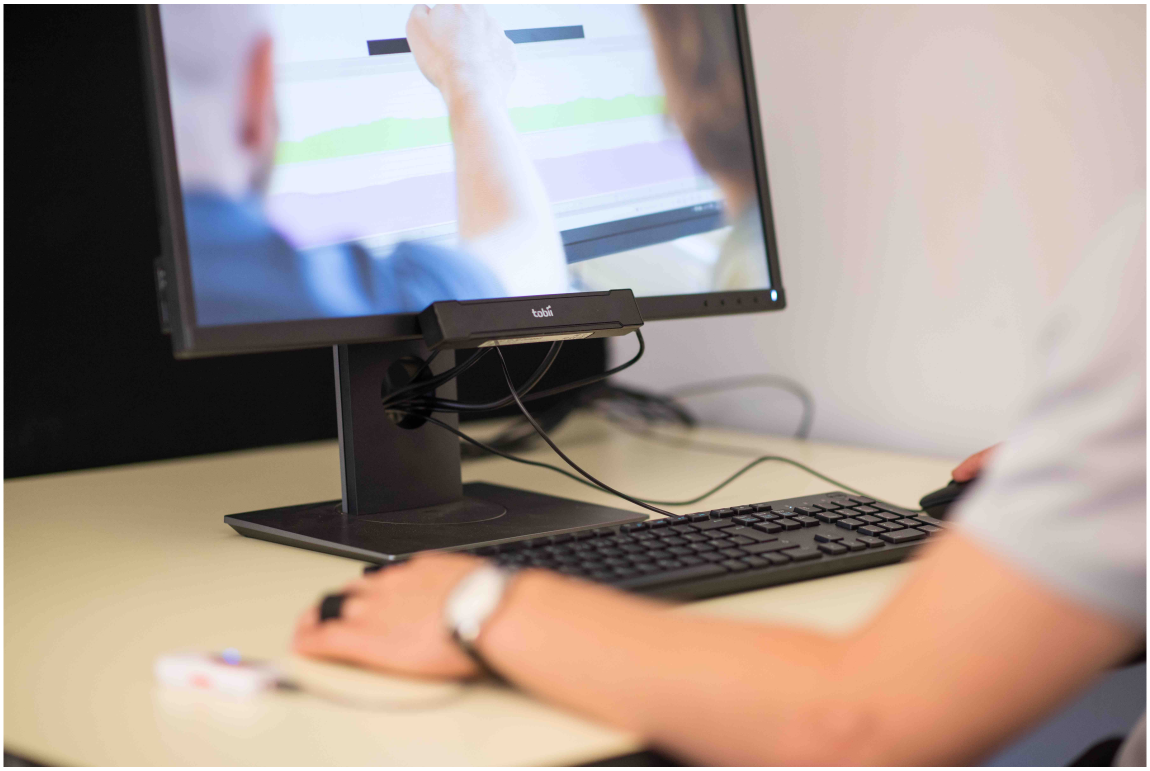

Figure 3.4: Remote eye tracker. Reprinted with permission from [5]

- Single-value features: For these features, a unique value is calculated for each data point by applying some model for all instances of the event (i.e. fixation, saccade or post-saccadic oscillation).

- Multi-value features: These are obtained by combining all instances of a feature into a distribution, which is then described using statistical characteristics such as mean, median, variance, skewness, and kurtosis.

Eye movement features are generally derived from the following:

- **Gaze point**: Gaze points constitute the basic unit of measure: one gaze point equals one raw sample captured by the eye tracker.

- **Fixations**: If a gaze point is maintained for a duration, it becomes a fixation, a period in which our eyes are locked towards a specific object. It can be thought of as the time periods when the eyes are relatively fixed on a single point of focus. During this period, the eye projects the point of interest on the retina. This is the point of sharpest vision. While in a fixation, eyes typically make small movements known as micro-saccades and ocular drifts [76]. (See Figure 3.5)

The features of interest from fixation are usually the duration and the rate. These are frequently used to analyze text comprehension and areas of interest in images. They have been used to study the subject's reading behavior, context, reading ability and age [77]. Other features such as velocity and acceleration of fixations are used for physiological studies and modeled as multi-value features.

- **Saccades**: The rapid eye movements used to change the point of fixations are generally referred to as saccades. When we read, for example, our eyes don't travel smoothly. They are locked in between letters (Figure 3.6). Separate from saccades, slower movements of the eye are referred to as *smooth-pursuit eye movement*. This type of movement is used to keep the eye aligned along with a moving stimulus or head movement. (See Figure 3.5)

  The duration and rate of saccades are the basic features of interest. Common features derived from these include amplitude, duration of a saccade, max speed, etc.

- **Post-Saccadic Oscillations**: Sometimes, eyes perform miniature oscillations at the end of a saccade, before fixating at a point. These are referred to as *Post-Saccadic Oscillations*. They help with overshoot correction and slower gliding movements known as *glissades*. The oscillations are correlated to the state of excitement and level of attention, while glissades are associated with vigilance [78]. They have also been associated with fatigue [79].

For a given system, the features used depend on the task being performed. Fixations are shown to be most important during image-based tasks, while saccades and other movements are used in video and text-based studies [56].

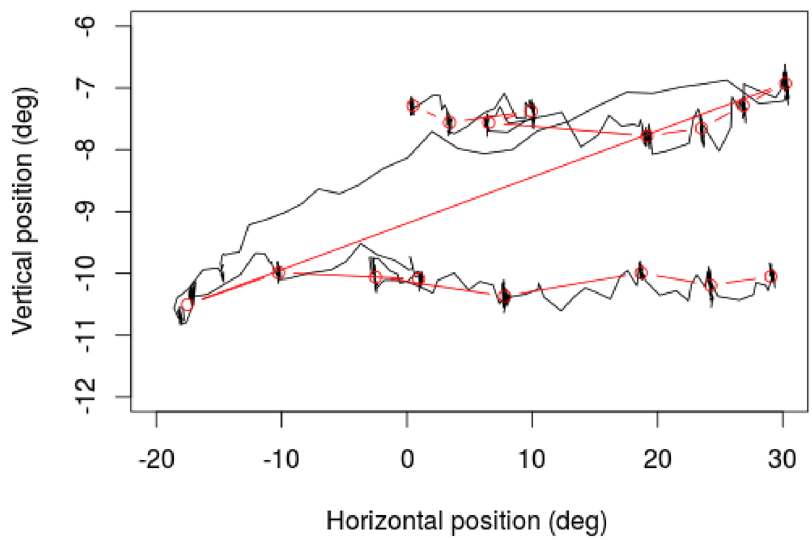

Figure 3.5: The red circles are Fixation points and the black lines connecting them are Saccades. Reprinted from [6]

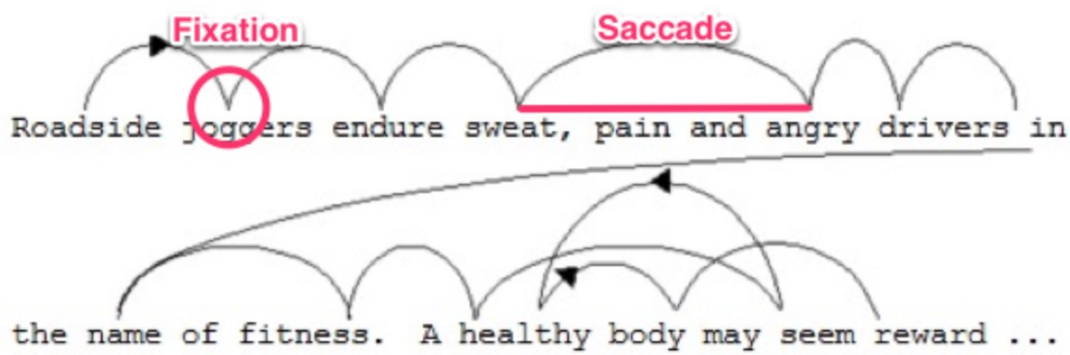

Figure 3.6: Saccades and Fixations when reading. Reprinted from [7]

4. METHODOLOGY

In this work, we collect eye movement data from multiple users and use various classification techniques to see how they compare to the baseline and one another. We use the FOVE VR headset (with built-in eye tracker) to collect data. We collect eye movement data by making all the users perform the same set of tasks.

4.1 FOVE VR Eye Tracker

We use the FOVE Virtual Reality Head Mounted Display (Figure 4.1) to capture eye movement data. It is light, easy to use VR headset that has two built-in infrared eye trackers with a capture framerate of 120 FPS and a tracking accuracy of less than 1 degree. Compared to other eye tracking VR headsets, it has lower system requirements making it viable to use on commodity general-purpose PCs. The presence of the eye tracker inside the headset makes it a good choice for eye-tracking applications in the VR domain. It also helps prevent loss of accuracy due to changes in head/body position. Also, the user calibrates just once and maintains the same distance from the eye-tracker at all times, even as they move their heads. It uses internal algorithms to determine the user's gaze in the VR scene and provides a Unity API for getting the gaze data.

4.2 Authentication Application

The VR authentication application was developed using Unity and the FOVE Unity API. Based on the issues with VR authentication methods proposed in [80, 4, 46] and limitations of legacy techniques, our system had the following requirements:

1. The system should be easy to use. Eye movement data should be collected with minimal additional effort.

2. It should be easy to integrate into existing VR applications.

Figure 4.1: FOVE headset. Reprinted from [8]

3. No physical movement should be needed for user identification.

4. It should be resistant to passcode spoofing and other forms of replication.

5. The time require for authentication should be short.

We collect user data using two different applications: One focused on smooth-pursuit eye movement and the other on rapid saccadic movement.

4.2.1 Method 1: Object Movement

In order to capture smooth-pursuit eye movement data and considering the requirements above, this method of authentication is a simple video stimulus with a moving object. The application shows a green circle on the screen, moving in predetermined fixed patterns. The user is required to follow the object with their eyes. This application captured smooth-pursuit eye movement data for basic movements, such as horizontal, vertical and diagonal lines (See Figure 4.2). As seen from the eye movement plot (See 4.3, even though the object moves along a diagonal, the resulting eye movement pattern is a more complex stroke or gesture.

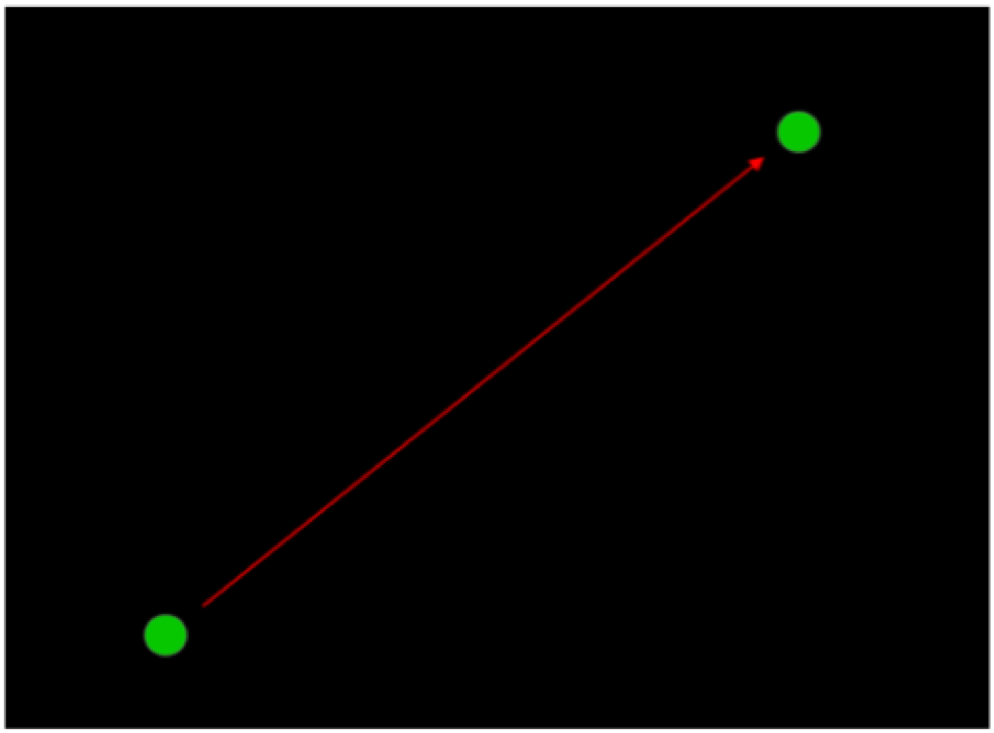

Figure 4.2: Moving object for smooth-pursuit eye movement data

As shown in Figure 4.3, the eye movement pattern resembles a complex sketch. Thus, this can be analyzed using sketch recognition based features and algorithms to generate useful information about the user.

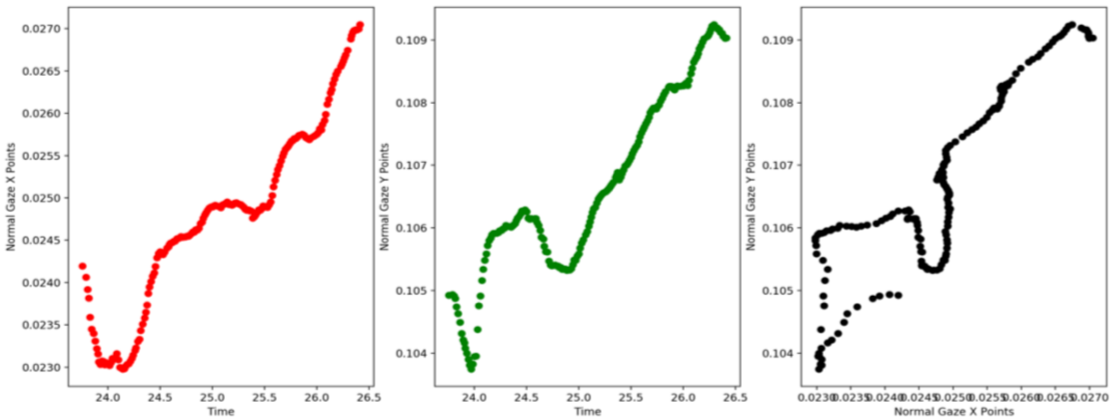

Figure 4.3: Eye movement pattern. (From left to right) a) Left eye vs time, b) Right eye vs time c) Left eye vs Right eye

4.2.1.1 *Object Movement as an Authentication Application*

Though we are using the above method for data collection, it can easily be used as an authentication application once a classification model has been trained. This method satisfies all the requirements mentioned above. As the stimulus is a simple object moving on the screen, the authentication process is short and easy. Since eye movement patterns are based on behavioral characteristics, they are hard to mechanically replicate making them secure to spoofing attacks. It is also easy to integrate such a system in any virtual reality based application. The effort required from the user is minimal. No physical movement is involved. One issue with this method is that it only relies on eye movement patterns for authentication. For increased security, systems combine multiple authentication methods [81, 82, 83, 84] and use a weighted combination of the results as the final output. Even with non-biometric identification systems, there is a rise in *Multi-factor Authentcation* systems [85, 86, 87, 88]. Although these systems provide additional layers of security, they require the user to perform authentication with multiple applications or devices which increases the cognitive load on the user [89]. There are methods being developed with easy to use biometrics such as voice [90].

Considering this, the ability to easily integrate into existing authentication systems is desirable, so that additional security can be added with minimal additional effort. The authentication task described above requires users to follow objects in a specific direction with minimal distraction. This is viable as a standalone method and has the advantage of simplicity and speed. Considering the advantages of multi-factor systems and ease of integration, we propose another approach which combines eye movement data with the well known and tested password-based authentication method.

4.2.2 Method 2: Gaze-based Passcode input

For ease of integration into existing methods and multi-factor authentication support, we propose a system that combines eye movement data with a password-based system. The method was developed in consideration of the lack of a haptic input device such as a keyboard for the VR environment. In this approach, we use a cursor controlled by the user's gaze. Numbers are selected on a virtual numpad to enter the passcode. See Figure 4.4. In order to provide feedback to the user of their selection, the application uses visual cues on the numpad buttons. The button lights up and changes colors from red to yellow to green, for a particular selection.

Instead of having a separate task for collecting eye movement data, the eye movement strokes are recorded while the user enters the passcode. With no additional effort from the user, the eye movement data is seamlessly collected. By combining the eye movement information with the passcode system, we add an additional layer of security in a completely transparent manner. Each movement from one digit of the passcode to another is captured as an eye movement stroke (See Figure 4.5). Thus, a passcode of length n produces $n-1$ strokes.

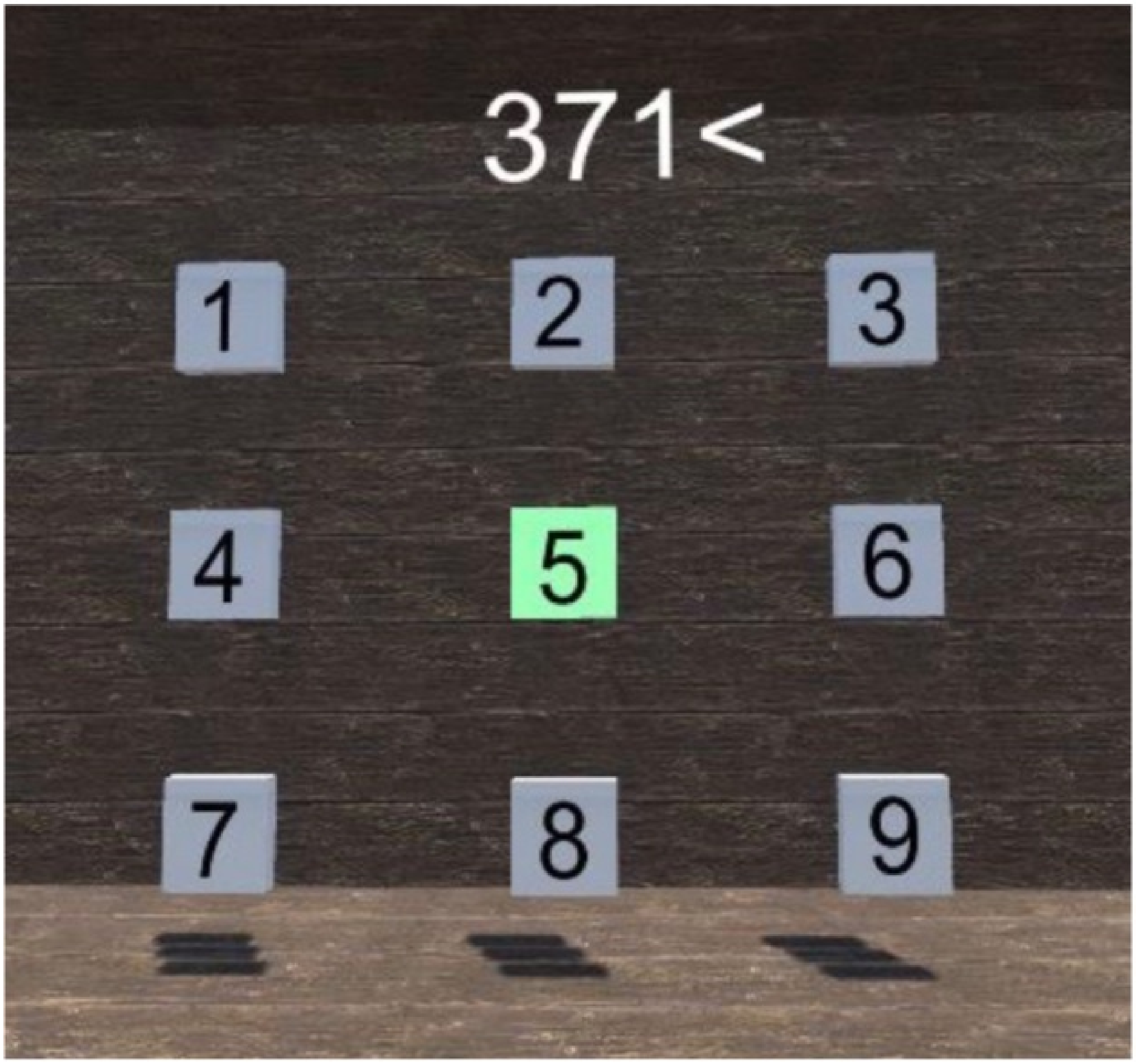

Figure 4.4: Gaze passcode input. Color changes are used to indicate selection and gaze feedback

4.2.2.1 Advantages of this Method

- **Multi-factor Authentication** There are two layers of security provided by this method: passcode and eye movement biometrics. This increases the overall security of the system. It is not just enough to guess the password. As seen before, replication of behavioral biometrics such as eye movement is hard.

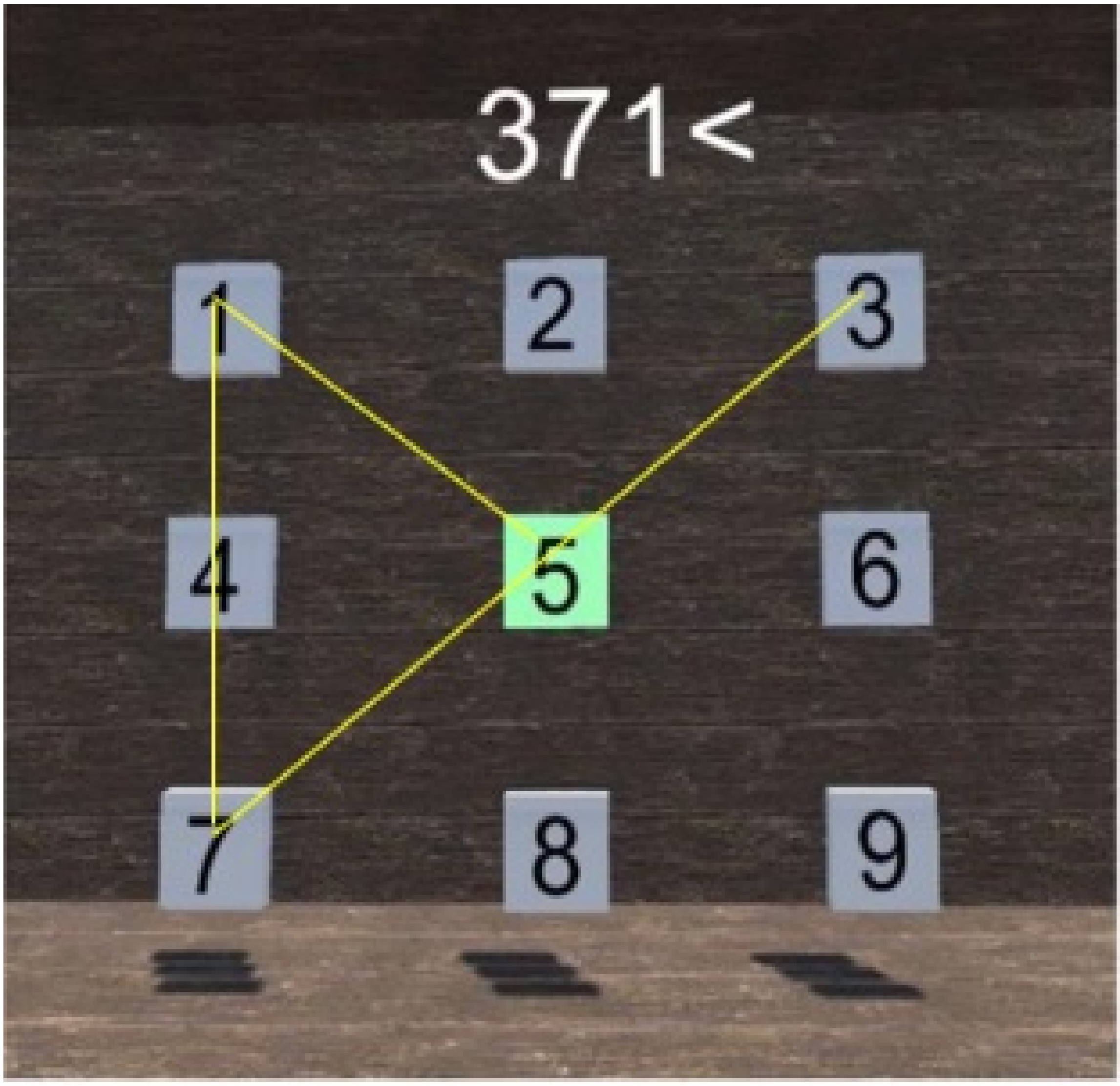

Figure 4.5: Three seperate eye movement strokes are obtained for a 4 digit passcode

- **Multiple strokes** For a given password of length n, we get $n-1$ strokes. In the previous task, we got one stroke per object movement. This allows us to combine multiple strokes and do the classification, which can prove useful. With high-frequency eye trackers, this will also provide us with more information on saccades and fixations. We can even measure the similarity between strokes.

5. IMPLEMENTATION

5.1 Data Preprocessing

For each stroke, the eye tracker starts and stops the data capture. As these rates could vary between different runs, it is important to preprocess the data to avoid uneven distribution of data points. Noise and fixations lead to an increase in density in the same region. This type of processing issues are well known in the sketch recognition [91, 92] and machine learning community [93, 94, 95].

Since we are treating the data points as a stroke for sketch recognition features and algorithms, we use the short-straw based method proposed in [91]. Resampling evens out the points by making the data points equidistant from each other based on a threshold. Figure 5.1 shows the difference in density before and after applying the resampling algorithm.

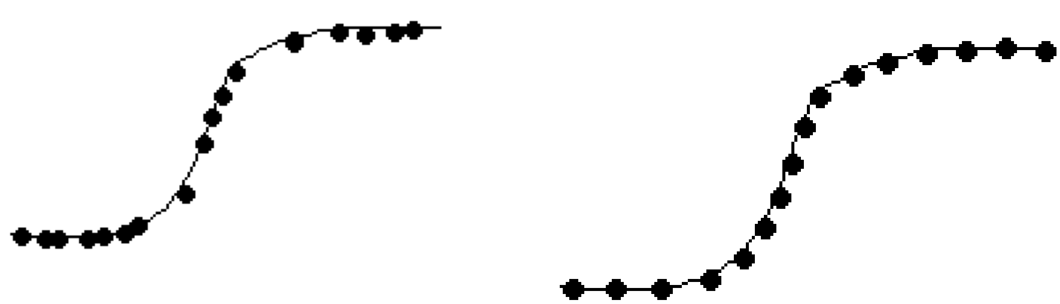

Figure 5.1: Resampling: raw data (left), resampled data (right)

This helps handle the edges of the stroke where a user selects the digit in method 2. The resampling algorithm is shown in Figure 5.2.

```
Data: A set P containing a series of N points, and an inter-point distance d_p
Result: A set R containing points which are at a distance d_p away from each other
begin
    R ⟵ {};
    // Start point for point pair in consideration
    startPoint ⟵ P[0];
    for i ∈ [1..N] do
        // Current Point in loop
        currentPoint = P[i];
        distance = Euclidean distance between currentPoint and startPoint;
        if distance >= d_p then
            // Find middle point between current point pair
            middlePoint ⟵ (0.0, 0.0);
            middlePoint.x = startPoint.x + (currentPoint.x - startPoint.x) * (d_p /
             distance);
            middlePoint.y = startPoint.y + (currentPoint.y - startPoint.y) * (d_p /
             distance);
            R = R ∪ middlePoint;
            startPoint = currentPoint;
        end
    end
    return R;
end
```

Figure 5.2: Resampling algorithm as described in [9]

The resampling algorithm was used for sketch features. For eye based features, the saccade and fixation extraction algorithm (IVT) handles these issues (explained below).

5.2 Feature Extraction

5.2.1 Sketch Features

As discussed before, we can treat each eye movement stroke as a sketch and hence extract sketch recognition based features for the same. There are different types of sketch recognition algorithms for recognizing sketches. Geometry-based algorithms [43] such as

Tahuti [42], PaleoSketch [11, 96, 97]] and Ladder [98, 99, 100, 101, 102, 103, 104, 105] are used as low level recognizers to identify primitive shapes in strokes. These can help enhance a users eye movement path.

The features we used for our system includes Rubine features [62, 106] and Paulson features [43]. The Rubine features were used in a gesture recognition system called GRANDMA. This system was used to create a gesture recognition application. As mentioned in [62], single-stroke gestures were used to interact with the system. The eye movement data in our application can be thought of as a single-stroke freehand sketch as the movement is continuous for a given stroke. We also included features from Paulson to capture the change in direction of saccades and velocity.

Some of Rubine and Paulson's features are shown in Figure 5.3. The features used for our eye movement analysis can be expressed as follows:

- *Features 1, 2: Cosine and Sine of starting angle of the stroke*

$$f_1 = cos(\alpha) = (x_2 - x_0)/\sqrt{[(y_2 - y_0)^2 + (x_2 - x_0)^2]}$$

$$f_2 = sin(\alpha) = (y_2 - y_0)/\sqrt{[(y_2 - y_0)^2 + (x_2 - x_0)^2]}$$

- *Feature 3: Length of the diagonal of the bounding box*

$$f_3 = \sqrt{(y_{max} - y_{min})^2 + (x_{max} - x_{min})^2}$$

(See Figure 5.3)

- *Feature 4: Angle of the diagonal of the bounding box*

$$f_4 = \arctan[(y_{max} - y_{min})/(x_{max} - x_{min}]$$

- *Feature 5: Total displacement*

$$f_5 = \sqrt{[(x_{n-1} - x_0)^2 + (y_{n-1} - y_0)^2]}$$

- *Feature 6, 7: Cosine and Sine of the angle between start and end points*

$$f_6 = cos(\beta) = (x_{n-1} - x_0)/f_5$$

$$f_7 = sin(\beta) = (y_{n-1} - y_0)/f_5$$

- *Feature 8: Total distance covered during movement (Stroke length)*

$$f_8 = \Sigma i = 1^{n-1}\sqrt{\Delta x_i^2 + \Delta y_i^2}$$

- *Feature 9, 10, 11: Total Rotation Measures*

  Rubine defines the angle between 2 lines formed by Δx_i and Δy_i as

$$\theta_i = \arctan(\frac{\Delta x_i * \Delta y_{i-1} - \Delta x_{i-1} * \Delta y_i}{\Delta x_i * \Delta x_{i-1} - \Delta y_{i-1} * \Delta y_i})$$

$$f_9 = \Sigma_{i=1}^{n-2}\theta_i$$

$$f_{10} = \Sigma_{i=1}^{n-2}|\theta_i|$$

$$f_{11} = \Sigma_{i=1}^{n-2}|\theta_i|^2$$

- *Feature 12: Max velocity during movement*

$$f_{12} = max_{i=1}^{n-1}(\Delta x_i^2 + \Delta y_i^2)/\Delta t_i^2$$

- *Feature 13, 14: Normalized Distance between Direction Extremes (NDDE), Direction Change Ratio (DCR)*

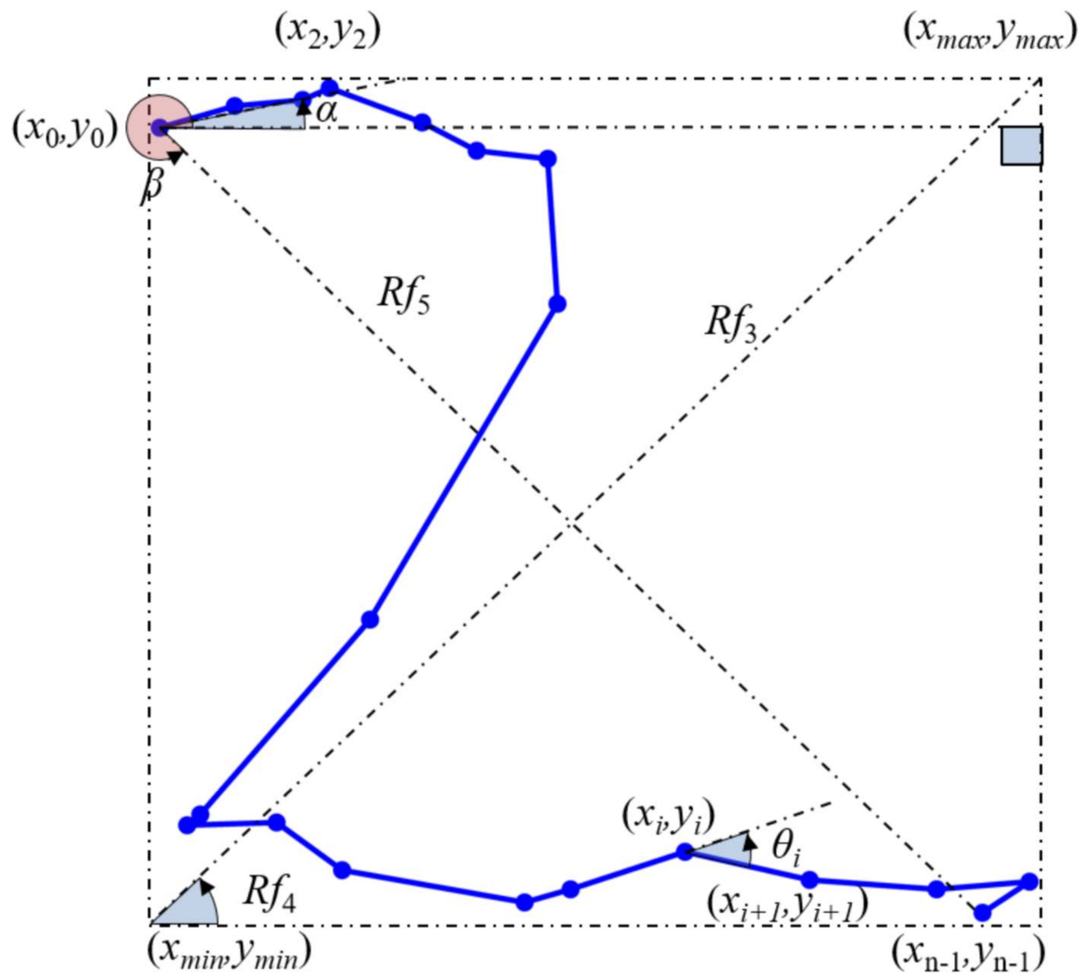

Figure 5.3: A visual reference for the terms defined in section 5.2.1. Reprinted from [10]

Each of the above-mentioned features captures the variation in saccadic movement and fixations of the eyes. Features 1 and 2 capture the start of a users saccade when they move from one point to another on the screen. This can vary based on saccadic oscillations. The bounding box features such as the angle of diagonal and length capture the size of the eye movement stroke. The Rotional measure based feature (9, 10, 11) is useful as different users could take a different amount of time for moving their eyes from one point to another. The Velocity of stroke is useful as it shows the max speed at which involuntary saccadic movements happen for a user (See Section 3.5). The last two features (13 and 14) capture the slow down and sharpness during direction changes [11]. To compute NDDR, we take the point with the highest direction value ($\Delta y_i/\Delta x_i$ and the point with lowest direction value and compute the stroke length between them. To computer DCR, by taking the maximum change in direction and divide it by the average change in direction. See Figure 5.4 for the type of strokes they capture.

5.3 Eye Movement Features

Based on fixations and saccades (described in Section 3.5), gaze data features were extracted. There are multiple algorithms for detecting fixations and saccades [107, 108, 109]. We used the simple *Identification by Velocity Threshold (IVT)* algorithm described in [109] for identifying fixations and saccades. More complex algorithms are useful when the data is collected using high-frequency eye trackers, as they contain more information. This algorithm calculates the velocity of movement at each data point and then classifies each point as a fixation or saccade based on a velocity threshold. Typically, angular velocities are used with thresholds $< 100°/sec$ for fixations and $< 300°/sec$ for saccades. But this can vary based on the study, eye tracker frequency, and implementation. Figure 5.5 shows the IVT algorithm. The saccade and fixation features used are described below.

Symbols and notations used for describing the various features are as follows:

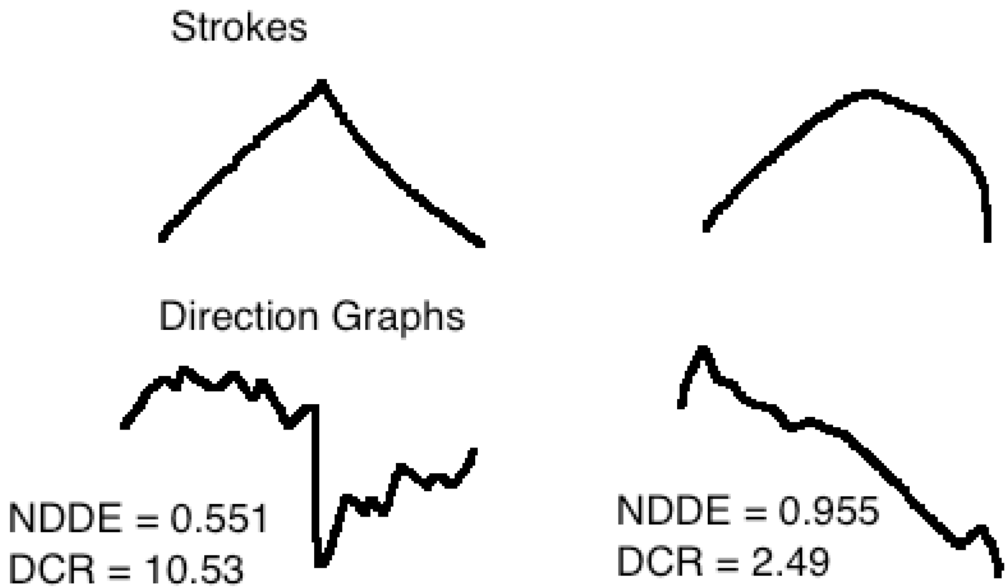

Figure 5.4: Direction graphs for 2 different strokes. Adapted from [11]

- Fix^{Num}, Sac^{Num}: Denotes total number of instances of an event-type in the data.
- $FixPos_i^j$, $SacPos_i^j$: Denotes the denote the *jth* positional sample of the *ith* instance of an event-type
- $FixVel_i(j)$, $SacVel_i(j)$: Denotes the *jth* velocity of the *ith* instance of an event-type

5.3.1 Fixation Features

Fixation features are used to define when the gaze is focused on a particular area of interest.

- *Features of temporal characteristics: Duration and Rate* These are the most basic of fixation features. They are defined as follows:

```
Data: A set V containing velocities at each datapoint, and a velocity threshold v_t
Result: Two sets F and S containing groups of fixation and saccade point indices
begin
    // Class:  fixation (0) or saccade (1)
    Classes ⟵ [for i ∈ [1..N], if(V[i] > v_t) then 0; else 1]
    F ⟵ {}, S ⟵ {}
    i = 0
    while i < N do
        // Current Group:  [i, j]
        j = GetPointWhereClassChanges(Classes, i)
        if Classes[i] == 0 then
            F.Add((i, j + 1))
        else
            S.Add((i, j + 1))
        end
        i = j + 1
    end
    return (F, S)
end
```

Figure 5.5: IVT algorithm

$$f1 = Fix_{rate} = Fix^{Num}/RecDur$$

where $RecDur$ is the total duration of the stroke or eye movement.

$$FixDur = \Sigma_{i=1}^{Fix^{Num}} FixDur_i$$

This captures the number of fixations and time spent in each fixation for a given stroke. Given that all users perform the same action in our study, it is useful in differentiating the users.

- *Total Fixation Duration* This is just the sum of the individual fixation durations. It captures time spent in fixation. It is usually combined with total time spent in stroke.

$$f2 = TotalFixDur = \Sigma_{i=1}^{FixNum} FixDur_i$$

- *Maximum Fixation Duration* This measures the fixation with maximum time.

$$f3 = MaxFixDur = max_{i=1}^{FixNum} FixDur_i$$

- *Fixation Displacement* This refers to the distance between the first and last point in a fixation.

$$f4 = FixDisp_i = EuclideanDist(FixEnd_i, FixStart_i)$$

- *Fixation Drift Distances* A simple way to model the fixation position is using the centroid of samples in the fixation profile. However, this does not capture the characteristics of a fixation drift. Fixation drift refers to the slow movement of eyes around a fixation point. This is an important feature as is determines how stable the user's fixations are.

  Fixation drift distances are computed using the distance between consecutive points that belong to a fixation.

$$f5 = FixDrift_i = \Sigma_{j=FixStart_i}^{FixEnd_i} EuclideanDist(P[j+1], P[j])$$

- *Maximum Fixation Displacement* This refers to the fixation with maximum displacement.

$$MaxFixDisp = max_i^N FixDisp_i$$

- *Maximum Fixation Drift* This refers to the least stable fixation. i.e. The one which drifts the most.

$$f7 = MaxFixDrift_i = max_i^N FixDrift_i$$

5.3.2 Saccade Features

As explained in Section 3.5, saccades are very fast movements of the eye from one point of focus to another. The peak velocities of saccades can reach over 600 °/s.

- *Features of temporal characteristics: Duration and Rate* These are the most basic of saccade features. They are defined as follows:

  $$f1 = Sac_{rate} = Sac^{Num}/RecDur$$

  where $RecDur$ is the total duration of the stroke or eye movement.

  $$SacDur = \Sigma_{i=1}^{Sac^{Num}} SacDur_i$$

  This captures the number of saccades and time spent moving from one fixation to another for a given stroke. Given that all users perform the same action in our study, it is useful in differentiating the users.

- *Total Saccade Duration* This is just the sum of the individual saccade durations. It captures time spent in moving between fixations through saccadic movement. It is usually combined with total time spent in stroke.

  $$f2 = TotalSacDur = \Sigma_{i=1}^{Sac^{Num}} SacDur_i$$

- *Maximum Saccade Duration* This measures the longest saccade by time.

$$f3 = MaxSacDur = max_{i=1}^{SacNum} SacDur_i$$

- *Saccade Amplitude* The amplitude of a saccade is usually used to describe its size.

$$f_4 = SacAmp_i = EuclideanDist(SacEnd_i, SacStart_i)$$

- *Saccade Travel Distances* This refers to the total distance traveled by a saccade. It is used in conjunction with Saccade Amplitude to understand the relative time spent in a saccade.

$$f_5 = SacTrvlDist_i = \Sigma_{j=SacStart_i}^{SacEnd_i} EuclideanDist(P[j+1], P[j])$$

- *Maximum Saccade Amplitude* This refers to the largest saccade amplitude.

$$f6 = MaxSacAmp = max_i^N SacAmp_i$$

- *Maximum Saccade Travel Distance* This refers to the longest saccade for the stroke.

$$MaxSacTrvlDist = max_i^N SacTrvlDist_i$$

More details on saccade and fixation features can be found in [76].

6. USER STUDY AND ANALYSIS

6.1 User Study

In order to understand useful features and perform classification, data needed to be collected from 20 users using both our authentication methods: Object Movement Stimulus (Section 4.2.1)and Gaze-based passcode input (Section 4.2.2). We describe the process of each activity. For both the studies, the user was wearing the FOVE0 VR headset (Figure 1.3) and sat in front of a computer running the application (See Figure 6.1).

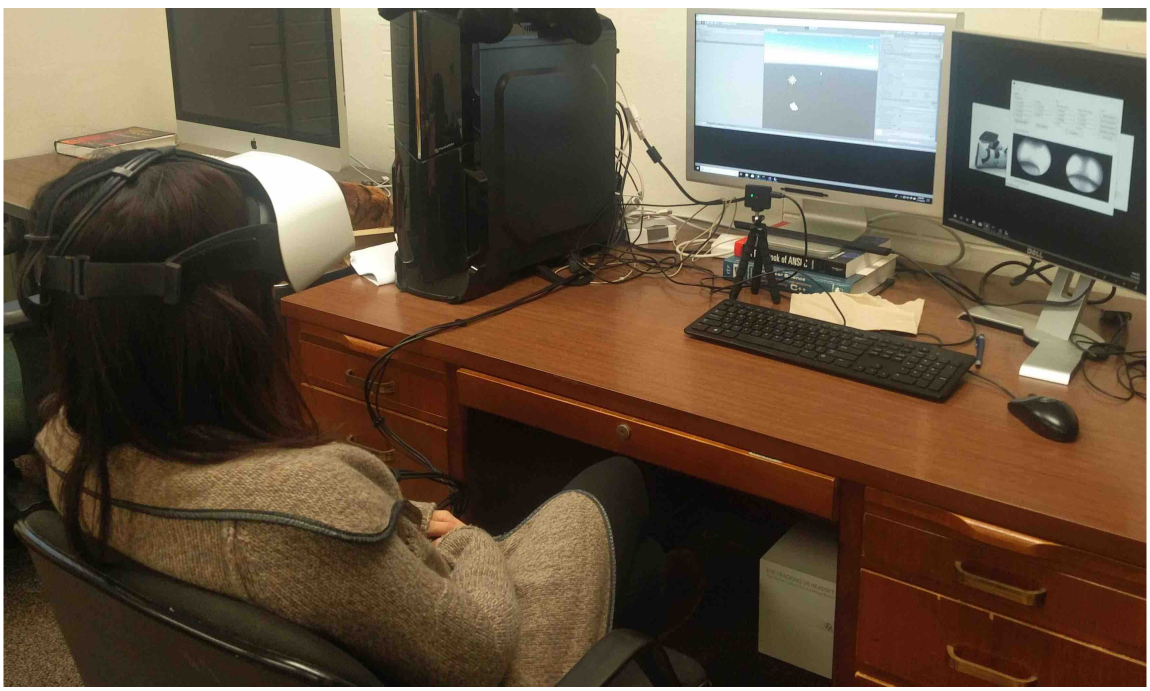

Figure 6.1: User study

6.1.1 Object Movement Stimulus

In this study, 20 participants were asked to follow a moving dot on the screen. The user was first calibrated for the eye tracker, after wearing the VR headset. The dot was a green circle in a black background as shown in Figure 4.2. The object traced out 2 patterns: Bottom Left to Top Right (Diagonal); and Left to Right (Horizontal). Each of the patterns was repeated 30 times with occasional breaks and recalibrations.

6.1.2 Gaze-based Passcode Input

In this study, 20 participants were asked to enter 2 passcodes (3794 and 1976) by gazing at the numbers (as explained in Section 4.2.2). Similar to the first task, calibration was done at the beginning and each passcode was repeated 30 times with occasional breaks and recalibrations.

6.2 Analysis

Before applying classification methods, we computed the various sketch, saccade and fixation features explained in Section 5.2. We had a total of 24 features (14 sketch features, 10 eye movement).

6.3 Subset Selection

An important aspect of machine learning based systems is to determine what kind of features to focus on. i.e. Discriminate between irrelevant and relevant parts of the data. Subset selection (or feature selection) refers to the process by which important features are selected, thus reducing the dimensions of the input data (See Figure 6.2). This helps reduce the computational cost and avoid overfitting.

The problem of feature selection can be expressed as follows:

- Given a set of features $f_1, f_2, ...f_n$, find a subset of features that "maximizes the classification algorithm's performance"

- The subset should maximize some given scoring function.

A large number of features can cause overfitting in the training set, thus reducing out of sample performance. Moreover, the computational effort is also high.

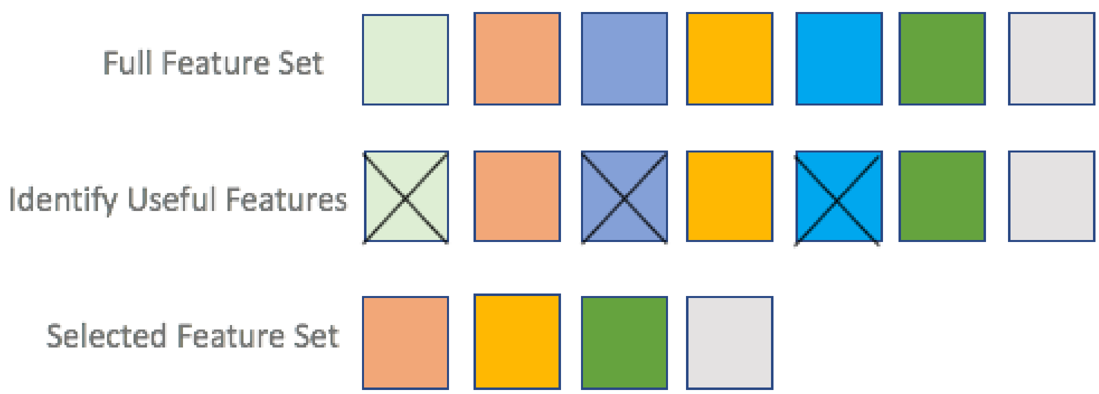

Figure 6.2: Example of feature selection. Adapted from [12]

6.3.1 Variable Ranking for Feature Selection

This method [110] usually uses a scoring function to rank the various features. The final subset of features is taken as the 'k' highest ranked values based on this scoring function. Though this method is not optimal, it is computationally efficient.

Ranking criteria are of two types:

- *Correlation based ranking* This score is calculated based on the correlation between a given feature and the target variable. The higher the correlation, the higher the score. Pearson Correlation Coefficient is used for computing the correlation and is given by:

$$R(f_i, y) = \frac{cov(f_i, y}{\sqrt{var(f_i var(y)}}$$

See Figure 6.3

Though this works well, it does not consider correlation/covariance between pairs of variables which can be significant. Feature diversity is also not considered.

- *Mutual Information based ranking* The main advantage of mutual information based ranking is that it can capture *non-linear dependencies*. It measures the information (entropy) shared between pairs of variables/features. But this is harder to estimate when compared to correlation.

- *Single Variable Classifier* This method performs the classification task with a single feature to measure its predictive power. Finally, a combination of these features (measured individually) is used to build the final model.

6.3.2 Feature Subset Selection

Feature subset selection aims to find the optimal subset of features. Broadly they can be classified into three types:

- **Filter Methods** Features are selected as a pre-processing step. This does not depend on the type of classifier used. Variable ranking (explained in Section 6.3.1) is a type of filter method. This is relatively easy to do and is the most commonly used method for subset selection (See Figure 6.4).

- **Wrapper Methods** In this case, the learner is treated as a black-box and the interface is to the learner is used for scoring various subsets. More details can be found in [111] (See Figure 6.5)

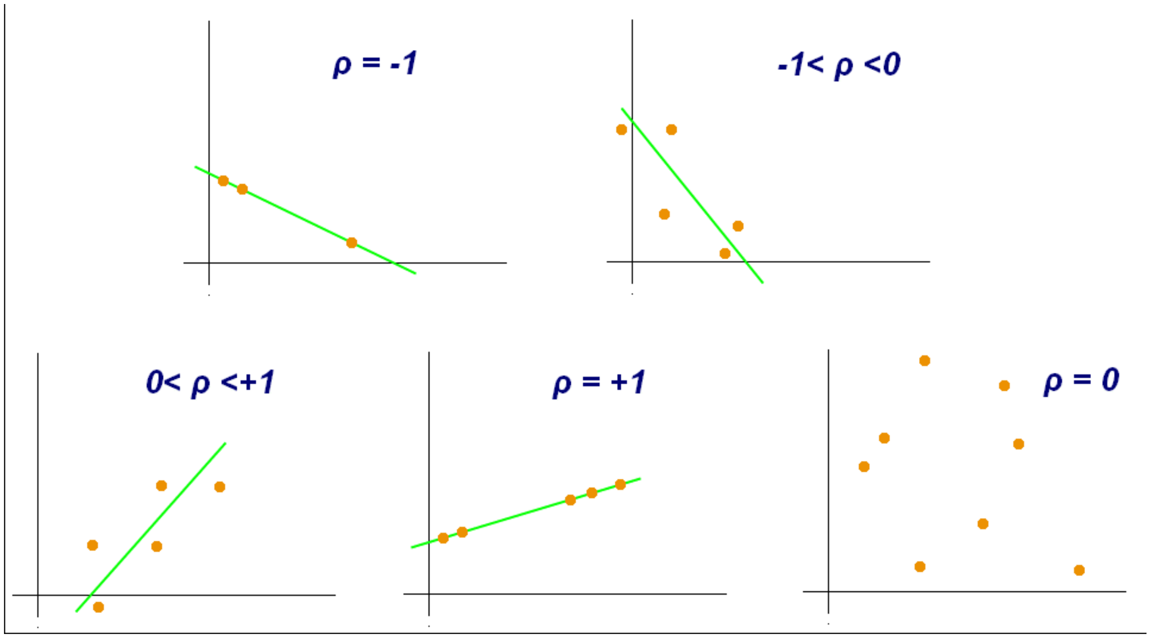

Figure 6.3: Values of correlation based on type of association. Reprinted from [13]

- **Embedded Methods** Embedded methods are specific to a given learning algorithm. Variable selection is performed in the process of training. See [112] for details.

We will use a filter based approach for feature selection. We divided the users randomly into a group of 15 for classification and 5 for feature selection. We use the Information Gain based attribute evaluator [113] to select features based on information gain (as described above). We also performed a correlation-based feature selection using the methods described below.

6.4 Feature Correlation

In order to perform subset selection, we first looked at all the features to understand the existence of linear relationships between them. This is done using a correlation matrix. Figure 6.6 shows that some features are correlated with others. Thus, it seemed like we could eliminate some features which captured the same information. In order to select a

Figure 6.4: Filter method for feature subset selection. Adapted from [12]

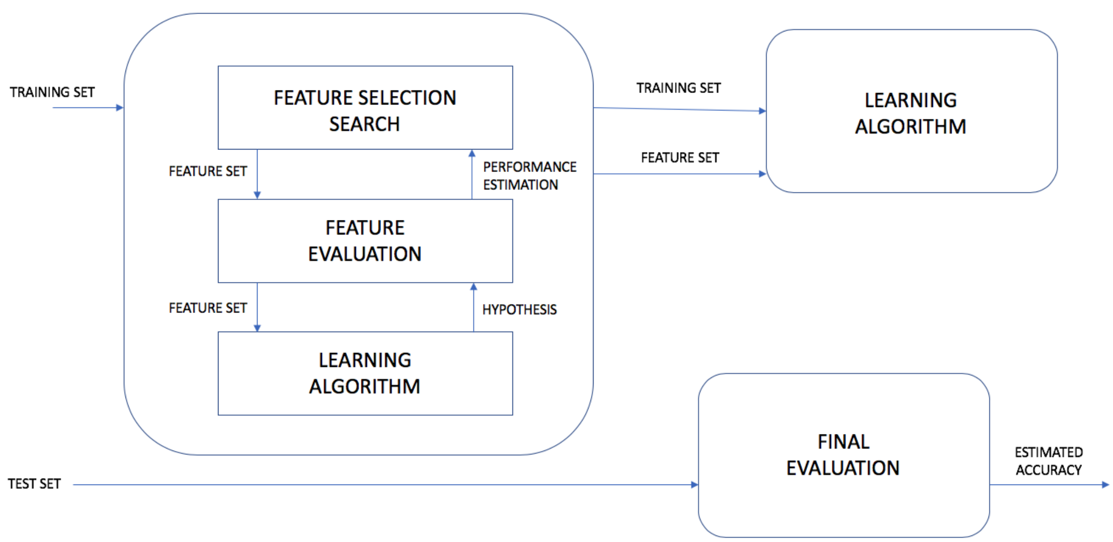

Figure 6.5: Wrapper method for feature subset selection. Adapted from [12]

subset of features based on correlation, we used two methods

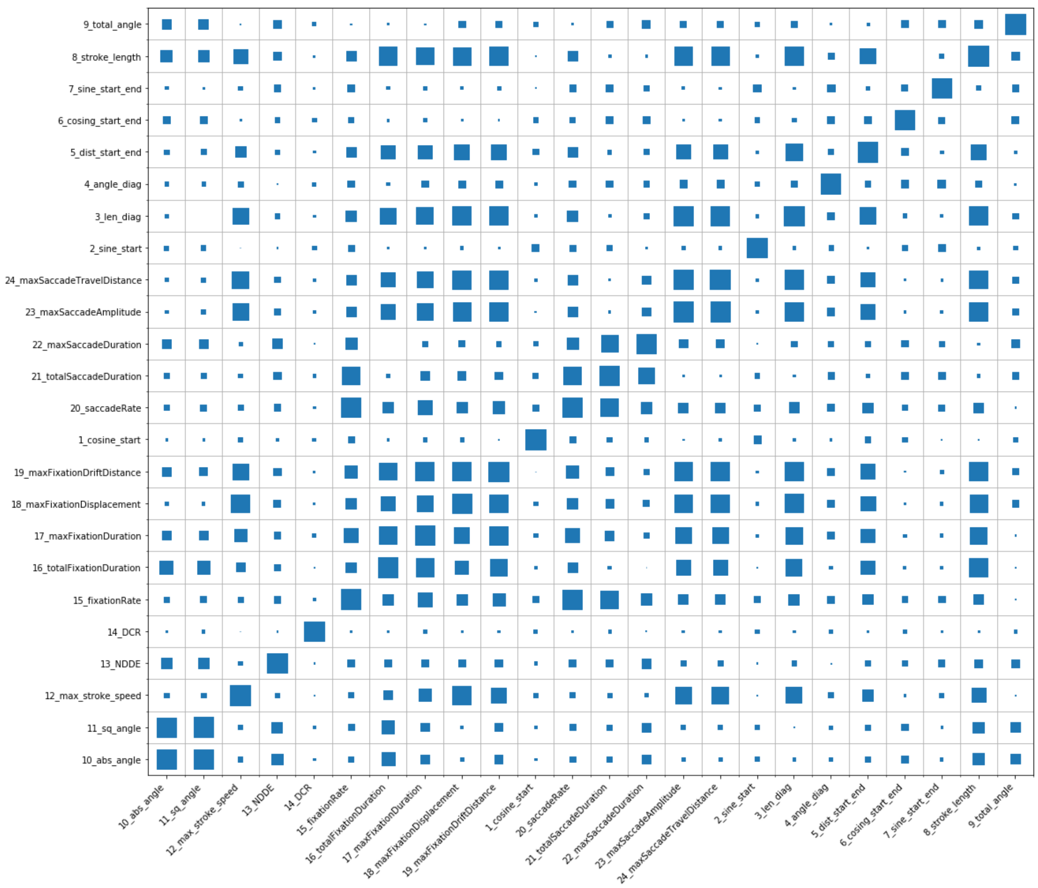

Figure 6.6: Correlation plot of features

- ANOVA Analysis
- Factor Analysis

Features	pvals	qvals
4_angle_diag	0.0	0.0
9_total_angle	0.0	0.0
12_max_stroke_speed	0.0	0.0
15_fixationRate	0.0	0.0
20_saccadeRate	0.0	0.0
24_maxSaccadeTravelDistance	0.0	0.0001
23_maxSaccadeAmplitude	0.0	0.0001
19_maxFixationDriftDistance	0.0001	0.0004
6_cosing_start_end	0.0006	0.0014
11_sq_angle	0.0006	0.0014
7_sine_start_end	0.0008	0.0017
10_abs_angle	0.0014	0.0028
17_maxFixationDuration	0.004	0.0072
18_maxFixationDisplacement	0.0042	0.0072
3_len_diag	0.0047	0.0074
13_NDDE	0.0488	0.0732
5_dist_start_end	0.0546	0.0771
8_stroke_length	0.0846	0.1074
21_totalSaccadeDuration	0.085	0.1074
14_DCR	0.1441	0.1729
2_sine_start	0.1762	0.2014
22_maxSaccadeDuration	0.3158	0.3445
16_totalFixationDuration	0.7405	0.7727
1_cosine_start	0.8104	0.8104

Figure 6.7: Significant features using ANOVA analysis

6.4.1 Anova Analysis

Analysis of variance (ANOVA) is a method used to analyze the differences in means among different groups. In our case, this helps us estimate features that are statistically significant in differentiating various users.

We used the python package *scipy* to perform the ANOVA analysis and estimate the associated p-values. Figure 6.7 shows the significant features.

In order to account for multiple tests, we performed FDR correction using the Benjamini-Hochberg procedure [114]. We obtained 15 significant features by using this procedure.

6.4.2 Factor Analysis

Factor Analysis [115] is used for finding underlying latent variables from a set of observed variables. It can be used for finding features that are related. By doing so, we can reduce the total of features in our classifier.

6.4.2.1 Adequacy Test

In order to perform factor analysis, it is common to evaluate the "sampling adequacy" of the dataset. We used the *Kaiser-Meyer-Olkin Test* [115] and obtained a value of 0.78, which indicates sampling adequacy.

6.4.2.2 Estimating Number of Factors

We performed Factor analysis using the *factor_analyzer* package in python. In order to determine the number of factors, we used a *scree plot* and computed the number of eigenvalues greater than 1. Figure 6.8 shows a scree plot of the eigenvalues. Using this, we estimate 7 factors.

Figure 6.9 shows the *loadings* obtained for the various factors. We used this information, along with significant values obtained from ANOVA to group similar factors and perform subset selection. 65% of the *Cumulative variance* was explained by the 5 factors.

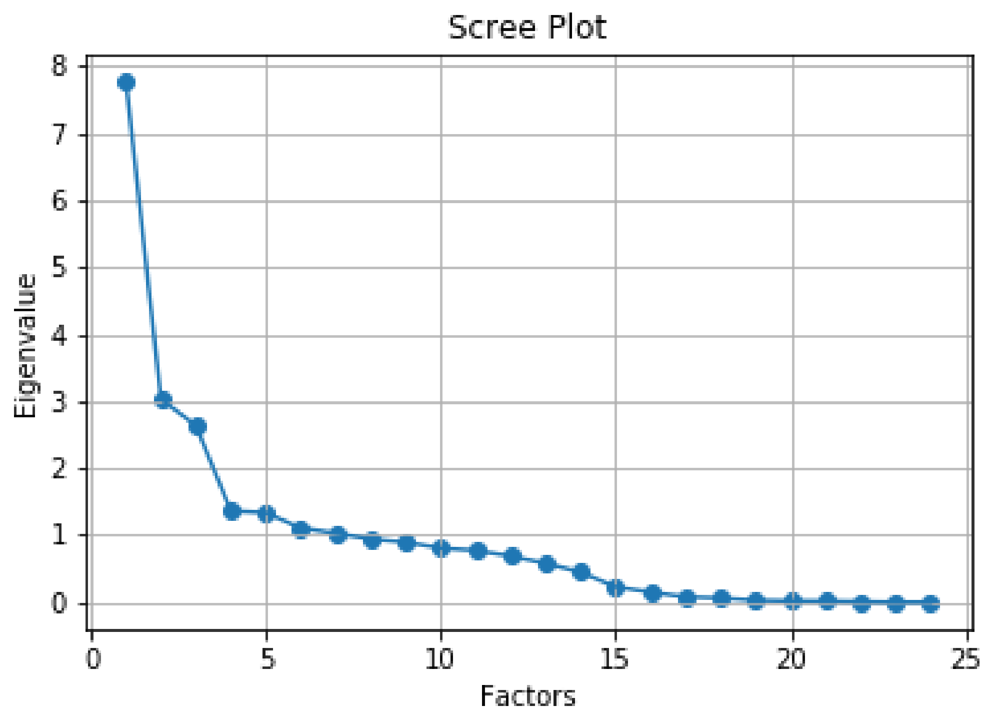

Figure 6.8: Scree plot of eigen values

Looking at the loading values, we can eliminate features that have high loading values for the same factor. For example, in Figure 6.9, we see that Fixation based features have high loading values for Factor 1. This helps us reduce the number of features.

6.5 User Classification

We used different types of classification algorithms that use different approaches. The following classifiers were used:

- **Bayesian Classifier** This classifier uses a probabilistic model to classify data based on the *Naive Bayes algorithm*. Naive Bayes is a probability-based machine learning algorithm that uses conditional probability and builds a Bayesian network.

- **J48 Decision Tree Classifier** Decision trees are one of the most commonly used ma-

Features	Factor_1	Factor_2	Factor_3	Factor_4	Factor_5	Factor_6	Factor_7
1_cosine_start	0.019	0.083	-0.015	-0.041	-0.073	0.152	-0.036
2_sine_start	0.012	-0.065	-0.055	0.037	0.185	-0.069	-0.019
3_len_diag	0.972	-0.05	-0.018	0.063	0.029	0.175	-0.071
4_angle_diag	0.106	-0.095	0.036	-0.031	0.268	0.002	-0.096
5_dist_start_end	0.59	-0.132	-0.11	0.279	0.015	0.539	-0.08
6_cosing_start_end	0.033	0.096	-0.156	0.061	-0.232	0.161	0.153
7_sine_start_end	-0.027	0.157	-0.049	-0.018	0.625	0.024	0.052
8_stroke_length	0.898	-0.036	0.302	0.156	-0.008	0.112	-0.007
9_total_angle	-0.066	0.07	-0.282	0.096	0.146	-0.023	0.318
10_abs_angle	0.049	0.077	0.98	0.084	0.032	-0.005	0.007
11_sq_angle	0.028	0.079	0.954	0.059	0.036	-0.002	0.036
12_max_stroke_speed	0.709	0.046	-0.091	-0.502	0.072	0.031	0.014
13_NDDE	-0.079	0.151	-0.334	0.112	0.111	0.101	0.073
14_DCR	-0.002	-0.011	0.005	-0.01	-0.039	-0.011	0.088
15_fixationRate	-0.257	0.897	0.063	-0.208	-0.074	0.16	0.008
16_totalFixationDuration	0.661	-0.077	0.404	0.425	0.017	0.081	0.361
17_maxFixationDuration	0.762	-0.295	0.172	0.221	0.066	-0.078	0.328
18_maxFixationDisplacement	0.901	-0.143	0.021	-0.314	0.044	0.163	-0.063
19_maxFixationDriftDistance	0.928	-0.184	0.161	0.011	0.032	0.022	0.148
20_saccadeRate	-0.251	0.902	0.068	-0.205	-0.076	0.16	0.014
21_totalSaccadeDuration	0.021	0.957	-0.02	0.247	-0.043	-0.006	-0.006
22_maxSaccadeDuration	0.116	0.579	-0.308	0.379	-0.004	-0.149	-0.083
23_maxSaccadeAmplitude	0.959	0.014	-0.072	0.012	0.017	-0.115	-0.17
24_maxSaccadeTravelDistance	0.97	0.021	-0.077	-0.047	0.021	-0.112	-0.162

Figure 6.9: Loading values for various factors

chine learning algorithms for many reasons. The output generated by decision trees is easy to understand. They try to interpret latent factors or relationships between the features. They implicitly perform feature selection based on the *gini impurity score* [116]. Trees are also simple to implement on smaller devices (with the less computational capability) as they are just a set of conditional statements.

- **Random Forest Classifier** Random Forest is an ensemble learning method for classification that uses many decision trees internally. It is a powerful classification method but requires a lot of memory and computation.

6.5.1 k-Fold Cross Validation

In order to divide the input dataset for training and testing, we used k-fold cross validation. Typically a value of 10 is chosen for *k*. In 10 fold cross-validation, the input dataset is randomly divided into 10 different groups of the same size. One group of 10% is used for testing and training is done with the other 90%. This process is repeated 10 times using a different 10% of the data for testing each time. Figure 6.10 shows this visually.

6.6 Classification Model

Using the above-mentioned classifiers and subset selection. We obtain the following results. Table 6.1 shows the output without any subset selection. We can see that All methods produce values significantly better than baseline. Random Forest does 8 times better than baseline estimates for both methods of input.

Table 6.1: Accuracy results for 20 users without subset selection

Classification Method	Object Movement	Passcode Based Input
Baseline (Random Chance)	5%	5%
Gaussian Naive Bayes	42.6%	25.5%
J48	30.93%	19.8%
Random Forest	49.6%	40.7%

Using the subset selection methods described above, we selected the following features

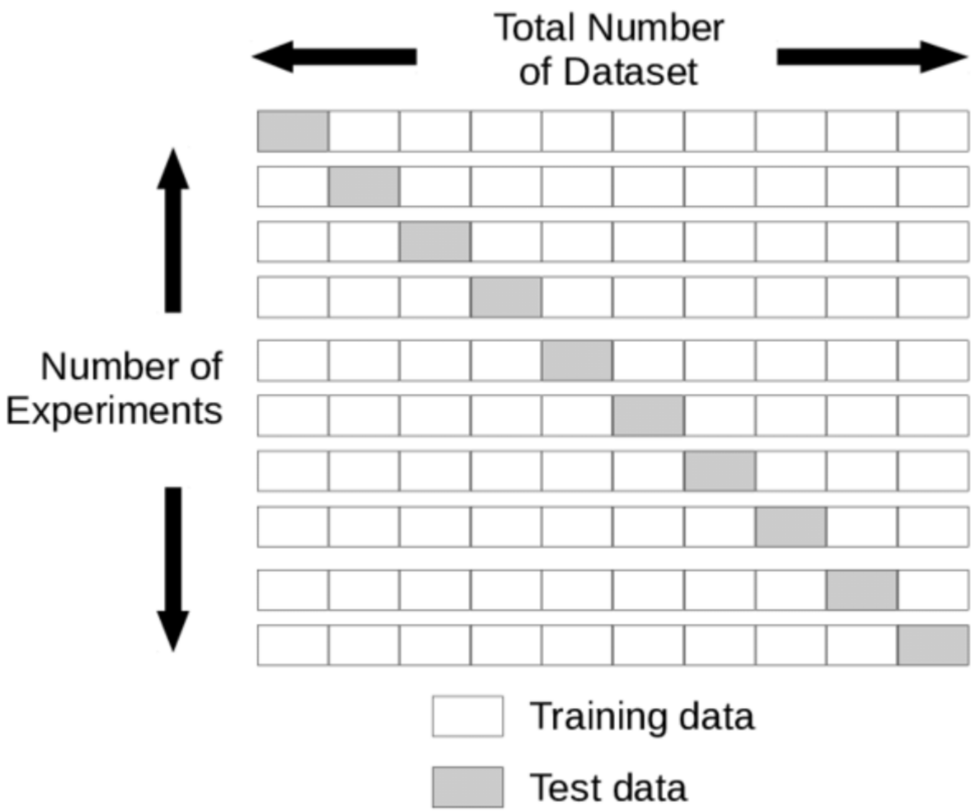

Figure 6.10: Example of 10 fold cross validation. Reprinted from [14]

from our total of 24 features. We computed the same metrics using the subset as shown in Table 6.3.

Table 6.2: Feature subset selected

Features Selected	
Total Angle	Max Stroke Speed
Stroke Length	Length of diagonal
Angle Diagonal	Absolute angle
Max Saccade Amplitude	Square Angle
Max Fixation Drift Distance	Angle of Diagonal

Table 6.3: Accuracy results with subset of 10 features for 15 users

Classification Method	**Object Movement**	**Passcode Based Input**
Baseline (Random Chance)	6%	6%
Gaussian Naive Bayes	38.6%	26.44%
J48	33.89%	20.2%
Random Forest	46.1%	35.7%

6.7 Discussion

As shown in Table 6.1, the accuracy for Random Forest in both methods is 8x better than baseline. This shows that there is unique information in eye movements that can be used for user identification. All of the data was collected using a 120hz eye tracker. With higher frequency eye trackers, we can collect more information detailed saccade and fixation information that can improve classification accuracy values.

- **Object Movement**: This method used a moving object stimulus. This is similar to other studies [20] that use a "jumping point" stimulus for identification. We can see that our subset of features captured most of the variation explained by all features. A possible reason for a slight decrease in accuracy could be that the input is non-linear and our subset selection algorithms use greedy subset selection methods. Our correlation based methods capture linear trends.

- **Gaze-based Passcode Input**: When compared to passcode based authentication, this system can significantly improve overall accuracy on top of the baseline. In practical systems, the classifier just needs to output each entry as genuine or spoofed. Such *Anomaly Detection* tasks can be simpler than multi-class classification as explained above. We used a single stroke from all our samples to perform the classification. It is possible to combine multiple strokes to build a more robust system in the future.

7. FUTURE WORK AND CONCLUSION

7.1 Future Work

7.1.1 Incorporate Physical Characteristics of Eye and Better Feature Extraction

The analysis and processing we did on the eye movement data was general, mainly due to the lower frequency of the data collected (120hz eye tracker). There are a few improvements that can be made in the feature extraction process, mainly for higher frequency data. Velocity and acceleration features are defined in [76]. These can provide more insights into saccade and fixation patterns of individuals. Frequency domain features known as Mel-frequency cepstral features are defined in [117], which can also be used. These can included as inputs to the classifier, along with sketch features optimized for the given activity. For example, use well known smooth-pursuit based features for the object movement task. Smooth pursuit eye movement are slower than saccadic movement and are usually used to keep the eyes aligned with a stimuli. Besides temporal characteristics, physical properties of the eye such as pupil size, etc. can be included. Modern VR headsets have add-ons that provide this information.

7.1.2 Combine with Other Biometrics

Keystroke and Eye-tracking biometrics were combined by [22] to achieve higher security levels. For example, IriTech provides an Iris Scanner OEM module which can be integrated into a VR headset. This can be easily combined with eye movement data and our authentication system to improve overall performance of the system.

7.1.3 One Shot Learning

The multi-class classification performed above shows the presence of uniqueness in eye movement. But in reality, a system just has to differentiate between the real user and malicious users. This means our system must be able to reject some users it has never seen before. This becomes an anomaly detection problem.

Another requirement for such systems is of onboarding a new user with few examples, instead of many trials. Such techniques are used in other authentication methods, such as face recognition and are implemented using *One-Shot Learning* [118, 119, 120].

The basic idea behind one-shot learning is to put similar data "closer" in a high dimensional space. This is done using a loss function called as *triplet loss*. This function takes three data samples, one anchor and two new samples and determines where the new sample is relative to the anchor. By doing so, with a few samples, the new user is put in a location in the higher dimensional space, relative to existing users. An example of such a system is Apple's FaceID, which uses just a few scans of facial expressions and creates a high-dimensional representation of the face. This is then placed along with other faces on which the system has already been trained.

7.2 Conclusion

In this work, we showed that eye movement biometrics can be used to augment existing authentication methods in the VR domain. As shown, their performance is found to be 8x better than the baseline (random guessing). This shows that there are inherent patterns in eye movement data that are unique to an individual. We developed features based on sketch recognition techniques. This allows us to use the work done in the domain of sketch recognition for eye tracking.

We proposed and built an authentication application for VR that can be easily integrated into existing VR applications. We built two interfaces, one for a simple object movement stimulus and the other was an eye movement augmented passcode system.

With the improvement of eye trackers and their VR integration, the input data will get better and our system. Moreover, with other biometric devices being integrated into VR (such as Retina scanners), our system can take additional biometric inputs and improve its overall performance.

REFERENCES

[1] Wikipedia, the free encyclopedia, "Virtual reality." https://bit.ly/1ZeZjAF, 2019. [Online; accessed April 27, 2019].

[2] Douglas G. Horner, O.D., Ph.D., Indiana University School of Optometry, "Oculomotor functions & neurology," 2004. http://www.opt.indiana.edu/v665/CD/CD_Version/CH4/CH4.HTM.

[3] FOVE, inc., "Fove headset." https://www.getfove.com/, 2019. [Online; accessed July 5, 2019].

[4] Z. Yu, H. Liang, C. Fleming, and K. L. Man, "An exploration of usable authentication mechanisms for virtual reality systems," in *2016 IEEE Asia Pacific Conference on Circuits and Systems (APCCAS)*, pp. 458–460, Oct 2016. https://doi.org/10.1109/APCCAS.2016.7804002.

[5] Bryn Farnsworth, "What is eye tracking and how does it work?," 2019. https://imotions.com/blog/eye-tracking-work.

[6] D. D. Salvucci and J. H. Goldberg, "Identifying fixations and saccades in eye-tracking protocols," in *Proceedings of the 2000 symposium on Eye tracking research & applications*, pp. 71–78, ACM, 2000. http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.68.2459&rep=rep1&type=pdf.

[7] Leighton's Blog, "How we read and why it matters," 2016. https://wldawson.wordpress.com/2016/03/03/how-we-read-and-why-it-matters/.

[8] Hugh Langley, "Fove heaset," 2016. https://www.wareable.com/vr/fove-review.

[9] A. Wolin, B. Eoff, and T. Hammond, "Shortstraw: A simple and effective corner finder for polylines," in *Proceedings of the Fifth Eurographics Conference on Sketch-Based In-*

terfaces and Modeling (SBIM), (Annecy, France, France), pp. 33–40, Eurographics Association, June 11-13, 2008. https://dl.acm.org/citation.cfm?id=2386301&picked=prox.

[10] F. Alamudun, "Analysis of visuo-cognitive behavior in screening mammography," PhD Doctoral Dissertation, Texas A&M University (TAMU), College Station, TX, USA, May 2016. Advisor: Tracy Hammond, ISBN: ORCID id: 0000-0002-0803-4542, http://hdl.handle.net/1969.1/157040.

[11] B. Paulson and T. Hammond, "Paleosketch: accurate primitive sketch recognition and beautification," in *Proceedings of the 13th international conference on Intelligent user interfaces*, pp. 1–10, ACM, 2008. http://doi.org/10.1145/1378773.1378775.

[12] Mehul Ved, "Feature selection and feature extraction in machine learning: An overview," 2018. https://bit.ly/2NrHKww.

[13] Wikipedia, the free encyclopedia, "Correlation," 2019. https://en.wikipedia.org/wiki/Pearson_correlation_coefficient#/media/File:Correlation_coefficient.png.

[14] A. Talpur, *Congestion Detection in Software Defined Networks using Machine Learning of Ali Murad Talpur*. PhD thesis, Universitat Bremen, 02 2017. https://doi.org/10.13140/RG.2.2.14985.85600.

[15] S. Choi, K. Jung, and S. D. Noh, "Virtual reality applications in manufacturing industries: Past research, present findings, and future directions," *Concurrent Engineering*, vol. 23, no. 1, pp. 40–63, 2015. https://doi.org/10.1177/1063293X14568814.

[16] Statista.com, "Vr users." https://www.statista.com/statistics/426469/active-virtual-reality-users-worldwide/, 2019. [Online; accessed April 27, 2019].

[17] S. Mandal, "Brief introduction of virtual reality & its challenges." https://bit.ly/2S0eGzM.

[18] A. George and A. Routray, “A score level fusion method for eye movement biometrics,” *Pattern Recognition Letters*, vol. 82, pp. 207 – 215, 2016. An insight on eye biometrics `http://www.sciencedirect.com/science/article/pii/S0167865515004067`.

[19] A. K. Jain, A. Ross, and S. Prabhakar, “An introduction to biometric recognition,” *IEEE Transactions on circuits and systems for video technology*, vol. 14, no. 1, pp. 4–20, 2004. `http://doi.org/10.1109/TCSVT.2003.818349`.

[20] L. Wang, *Behavioral Biometrics for Human Identification: Intelligent Applications: Intelligent Applications*. IGI Global, 2009. `https://dl.acm.org/citation.cfm?id=1803843`.

[21] M. A. Just and P. A. Carpenter, “Eye fixations and cognitive processes,” *Cognitive Psychology*, vol. 8, no. 4, pp. 441 – 480, 1976. `http://www.sciencedirect.com/science/article/pii/0010028576900153`.

[22] D. Silver and A. Biggs, “Keystroke and eye-tracking biometrics for user identification.,” in *Keystroke and Eye-Tracking Biometrics for User Identification*, pp. 344–348, 01 2006. `https://bit.ly/2FZjLUf`.

[23] R. Carpenter, “The saccadic system: A neurological microcosm,” *Advances in Clinical Neuroscience and Rehabilitation*, vol. 4, pp. 6–8, 01 2004. `http://www.acnr.co.uk/pdfs/volume4issue1/v4i1reviewart1.pdf`.

[24] R. Leigh and D. Zee, *The Neurology of Eye Movements*. Contemporary neurology series, Oxford University Press, 2015. `https://books.google.com/books?id=v2s0BwAAQBAJ`.

[25] G. Bargary, J. M. Bosten, P. T. Goodbourn, A. J. Lawrance-Owen, R. E. Hogg, and J. D. Mollon, “Individual differences in human eye movements: An oculomotor signature?,” *Vision Res.*, vol. 141, pp. 157–169, 12 2017. `https://doi.org/10.1016/j.visres.2017.03.001`.

[26] I. MeyhÃűfer, K. Bertsch, M. Esser, and U. Ettinger, "Variance in saccadic eye movements reflects stable traits," *Psychophysiology*, vol. 53, no. 4, pp. 566–578, 2016. https://onlinelibrary.wiley.com/doi/abs/10.1111/psyp.12592.

[27] J. Katsanis, J. Taylor, W. G. Iacono, and M. A. Hammer, "Heritability of different measures of smooth pursuit eye tracking dysfunction: a study of normal twins," *Psychophysiology*, vol. 37, pp. 724–730, Nov 2000. https://doi.org/10.1111/1469-8986.3760724.

[28] C. Klein and B. Fischer, "Instrumental and test-retest reliability of saccadic measures," *Biol Psychol*, vol. 68, pp. 201–213, Mar 2005. https://doi.org/10.1016/j.biopsycho.2004.06.005.

[29] I. Meyhofer, K. Bertsch, M. Esser, and U. Ettinger, "Variance in saccadic eye movements reflects stable traits," *Psychophysiology*, vol. 53, pp. 566–578, Apr 2016. http://doi.org/10.1111/psyp.12592.

[30] N. Smyrnis, "Metric issues in the study of eye movements in psychiatry," *Brain Cogn*, vol. 68, pp. 341–358, Dec 2008. https://doi.org/10.1016/j.bandc.2008.08.022.

[31] G. Vikesdal and T. Langaas, "Saccade latency and fixation stability: Repeatability and reliability," *Journal of Eye Movement Research*, vol. 9, 01 2016. https://doi.org/10.16910/jemr.9.2.3.

[32] P. Kasprowski and J. Ober, "Eye movements in biometrics," in *Eye Movements in Biometrics*, pp. 248–258, 05 2004. https://doi.org/10.1007/978-3-540-25976-3_23.

[33] O. Komogortsev, A. Karpov, and C. Holland, "Attack of mechanical replicas: Liveness detection with eye movements," *IEEE Transactions on Information Forensics and Security*, vol. 10, pp. 716–725, 02 2015. https://doi.org/10.1109/TIFS.2015.2405345.

[34] A. Kumar and G. Krol, "Binocular infrared oculography," *The Laryngoscope*, vol. 102, no. 4, pp. 367–378, 1992. https://onlinelibrary.wiley.com/doi/abs/10.1288/00005537-199204000-00002.

[35] Y. Kong, H. Lee, N. Kim, S. Lee, J. Park, T. Han, and Y. Nam, "Low-cost infrared video-oculography for measuring rapid eye movements," in *Advanced Multimedia and Ubiquitous Engineering* (J. J. J. H. Park, S.-C. Chen, and K.-K. Raymond Choo, eds.), (Singapore), pp. 258–262, Springer Singapore, 2017. https://doi.org/10.1007/978-981-10-5041-1_44.

[36] T. Hammond, "Learning through the lens of sketch," in *Frontiers in Pen and Touch: Impact of Pen and Touch Technology on Education* (A. A. T. Hammond and M. Prasad, eds.), Human-Computer Interaction Series, ch. 21, pp. 301–341, Switzerland: Springer, 2017. ISBN: 978-3-319-64239-0, https://doi.org/10.1007/978-3-319-64239-0_21.

[37] C. J. Alvarado, *A natural sketching environment: Bringing the computer into early stages of mechanical design*. PhD thesis, Massachusetts Institute of Technology, 2000. https://bit.ly/2NAA6Vv.

[38] T. M. Sezgin, T. Stahovich, and R. Davis, "Sketch based interfaces: early processing for sketch understanding," in *Proceedings of the 2001 workshop on Perceptive user interfaces*, pp. 1–8, ACM, 2001. https://dl.acm.org/citation.cfm?id=971478.

[39] T. Hammond, "Enabling instructors to develop sketch recognition applications for the classroom," in *2007 37th Annual Frontiers In Education Conference - Global Engineering: Knowledge Without Borders, Opportunities Without Passports*, pp. S3J–11–S3J–16, Oct 2007. https://doi.org/10.1109/FIE.2007.4417930.

[40] T. Hammond, "Workshop - integrating sketch recognition technologies into your classroom," in *2008 38th Annual Frontiers in Education Conference*, pp. W2C–1–W2C–1, Oct 2008. https://doi.org/10.1109/FIE.2008.4720505.

[41] T. Hammond, "Dialectical creativity: Sketch-negate-create," in *Studying Visual and Spatial Reasoning for Design Creativity* (J. S. Gero, ed.), (Dordrecht), pp. 91–108, Springer Netherlands, 2015. https://doi.org/10.1007/978-94-017-9297-4.

[42] T. Hammond and R. Davis, "Tahuti: A geometrical sketch recognition system for uml class diagrams," in *Technical Report SS-02-08: Papers from the 2002 Association for the Advancement of Artificial Intelligence (AAAI) Spring Symposium on Sketch Understanding*, (Menlo Park, California, USA), p. 8 pages, AAAI, July 28–August 1, 2002. https://dl.acm.org/citation.cfm?id=1185786.

[43] B. Paulson, P. Rajan, P. Davalos, R. Gutierrez-Osuna, and T. Hammond, "What!?! no rubine features?: using geometric-based features to produce normalized confidence values for sketch recognition," in "", 2008. https://bit.ly/2G0DUZY.

[44] H. Choi and T. Hammond, "Sketch recognition based on manifold learning," in *Proceedings of the 23rd National Conference on Artificial Intelligence - Volume 3*, AAAI'08, pp. 1786–1787, AAAI Press, 2008. http://dl.acm.org/citation.cfm?id=1620270.1620353.

[45] H. Choi, B. Paulson, and T. Hammond, "Gesture recognition based on manifold learning," in *Structural, Syntactic, and Statistical Pattern Recognition (SSPR), Lecture Notes in Computer Science*, vol. 5342, pp. 247–256, Springer Berlin Heidelberg, 2008. https://doi.org/10.1007/b98738.

[46] R. Dasika and S. Tiruvayipati, "A more secure authentication through a simple virtual environment," *International Journal of Advanced Computer Research*, vol. 2, pp. 113–116, 06 2011. https://bit.ly/2JjdAfJ.

[47] F. A. Alsulaiman and A. El Saddik, "Three-dimensional password for more secure authentication," *IEEE Transactions on Instrumentation and Measurement*, vol. 57, pp. 1929–1938, Sep. 2008. https://doi.org/10.1109/TIM.2008.919905.

[48] Y. Shen, H. Wen, C. Luo, W. Xu, T. Zhang, W. Hu, and D. Rus, “Gaitlock: Protect virtual and augmented reality headsets using gait,” *IEEE Transactions on Dependable and Secure Computing*, vol. 16, pp. 484–497, May 2019. `https://doi.org/10.1109/TDSC.2018.2800048`.

[49] J. Weaver, K. Mock, and B. Hoanca, “Gaze-based password authentication through automatic clustering of gaze points,” in *2011 IEEE International Conference on Systems, Man, and Cybernetics*, pp. 2749–2754, Oct 2011. `10.1109/ICSMC.2011.6084072`.

[50] V. Rajanna, S. Polsley, P. Taele, and T. Hammond, “A gaze gesture-based user authentication system to counter shoulder-surfing attacks,” in *Proceedings of the 2017 CHI Conference Extended Abstracts on Human Factors in Computing Systems*, CHI EA ’17, (New York, NY, USA), pp. 1978–1986, ACM, 2017. `http://doi.acm.org/10.1145/3027063.3053070`.

[51] P. Kasprowski and J. Ober, “Eye movements in biometrics,” in *Biometric Authentication* (D. Maltoni and A. K. Jain, eds.), (Berlin, Heidelberg), pp. 248–258, Springer Berlin Heidelberg, 2004. `http://doi.org/10.1007/978-3-540-25976-3_23`.

[52] R. Bednarik, T. Kinnunen, A. Mihaila, and P. FrÃd’nti, “Eye-movements as a biometric,” 2005. `https://doi.org/10.1007/11499145_79`.

[53] T. Kinnunen, F. Sedlak, and R. Bednarik, “Towards task-independent person authentication using eye movement signals,” in *Proceedings of the 2010 Symposium on Eye-Tracking Research & Applications*, ETRA ’10, (New York, NY, USA), pp. 187–190, ACM, 2010. `http://doi.acm.org/10.1145/1743666.1743712`.

[54] I. Rigas, G. Economou, and S. Fotopoulos, “Biometric identification based on the eye movements and graph matching techniques,” *Pattern Recogn. Lett.*, vol. 33, pp. 786–792, Apr. 2012. `http://dx.doi.org/10.1016/j.patrec.2012.01.003`.

[55] I. Rigas, G. Economou, and S. Fotopoulos, “Human eye movements as a trait for biometrical identification,” in *2012 IEEE Fifth International Conference on Biometrics: Theory,*

Applications and Systems (BTAS), pp. 217–222, Sep. 2012. https://doi.org/10.1109/BTAS.2012.6374580.

[56] U. Saeed, "Eye movements during scene understanding for biometric identification," *Pattern Recogn. Lett.*, vol. 82, pp. 190–195, Oct. 2016. https://doi.org/10.1016/j.patrec.2015.06.019.

[57] V. Rajanna and T. Hammond, "Gaze-assisted user authentication to counter shoulder-surfing attacks," *CoRR*, vol. abs/1803.07782, 2018. https://dblp.org/rec/bib/journals/corr/abs-1803-07782.

[58] V. Rajanna, S. Polsley, P. Taele, and T. Hammond, "A gaze gesture-based user authentication system to counter shoulder-surfing attacks," in *Proceedings of the 2017 CHI Conference Extended Abstracts on Human Factors in Computing Systems*, CHI EA '17, (New York, NY, USA), pp. 1978–1986, ACM, 2017. http://doi.acm.org/10.1145/3027063.3053070.

[59] A. De Luca, M. Denzel, and H. Hussmann, "Look into my eyes!: Can you guess my password?," in *Proceedings of the 5th Symposium on Usable Privacy and Security*, p. 7, ACM, 2009. https://bit.ly/2S3qj9d.

[60] V. Cantoni, C. Galdi, M. Nappi, M. Porta, and D. Riccio, "Gant: Gaze analysis technique for human identification," *Pattern Recognition*, vol. 48, no. 4, pp. 1027–1038, 2015. https://doi.org/10.1016/j.patcog.2014.02.017.

[61] Z. Liang, F. Tan, and Z. Chi, "Video-based biometric identification using eye tracking technique," in *2012 IEEE International Conference on Signal Processing, Communication and Computing (ICSPCC 2012)*, pp. 728–733, Aug 2012. https://doi.org/10.1109/ICSPCC.2012.6335584.

[62] D. Rubine, "Specifying gestures by example," in *"ACM SIGGRAPH Computer Graphics"*, vol. 25, pp. 329–337, 07 1991. https://doi.org/10.1145/122718.122753.

[63] H.-J. Yoon, F. Alamudun, K. Hudson, G. Morin-Ducote, and G. Tourassi, "Deep gaze velocity analysis during mammographic reading for biometric identification of radiologists," *Journal of Human Performance in Extreme Environments*, vol. 14, no. 1, p. 3, 2018. `https://doi.org/10.7771/2327-2937.1088`.

[64] H.-J. Y. G. D. T. Folami Alamudun, Paige F. Paulus, "Modeling sequential context effects in diagnostic interpretation of screening mammograms," *Journal of Medical Imaging*, vol. 5, no. 3, pp. 1 – 7 – 7, 2018. `https://doi.org/10.1117/1.JMI.5.3.031408`.

[65] F. Alamudun, H.-J. Yoon, K. B. Hudson, G. Morin-Ducote, T. Hammond, and G. D. Tourassi, "Fractal analysis of visual search activity for mass detection during mammographic screening," *Medical Physics*, vol. 44, no. 3, pp. 832–846, 2017. `https://aapm.onlinelibrary.wiley.com/doi/abs/10.1002/mp.12100`.

[66] F. Alamudun, H.-J. Yoon, T. Hammond, K. Hudson, G. Morin-Ducote, and G. Tourassi, "Shapelet analysis of pupil dilation for modeling visuo-cognitive behavior in screening mammography," in "", p. 97870M, 03 2016. `https://doi.org/10.1117/12.2217670`.

[67] V. Navalpakkam and E. F. Churchill, *Eye Tracking: A Brief Introduction*, pp. 323–348. New York, NY: Springer New York, 2014. `https://doi.org/10.1007/978-1-4939-0378-8_13`.

[68] "Methods of measuring eye movements," 2019. `https://www.liverpool.ac.uk/~pcknox/teaching/Eymovs/emeth.htm`.

[69] S. Gulyás, *Electro-Oculography (EOG) Examination of Eye Movements*, pp. 287–293. Cham: Springer International Publishing, 2016. `https://doi.org/10.1007/978-3-319-28956-4_33`.

[70] C. Jian-nan, Z. Peng-yi, Z. Si-yi, Z. Chuang, and H. Ying, "Key techniques of eye gaze tracking based on pupil corneal reflection," in *2009 WRI Global Congress on Intelligent Sys-*

tems, vol. 2, pp. 133–138, IEEE, 2009. https://doi.org/10.1109/GCIS.2009.338.

[71] N. M. Esfahani, "A brief review of human identification using eye movement," *Journal of Pattern Recognition Research*, vol. 1, pp. 15–24, 2016. https://doi.org/10.13176/11.705.

[72] J. E. Richards and S. K. Hunter, "Testing neural models of the development of infant visual attention," *Developmental Psychobiology: The Journal of the International Society for Developmental Psychobiology*, vol. 40, no. 3, pp. 226–236, 2002. https://bit.ly/2xw6nT0.

[73] B. Bauman, R. Gunhouse, A. Jones, W. D. Silva, S. Sharar, V. Rajanna, J. Cherian, J. I. Koh, and T. A. Hammond, "Visualeyeze: A web-based solution for receiving feedback on artworks through eye-tracking," in *IUI Workshops*, 2018. https://bit.ly/2Jjuhrs.

[74] K. Kaspar and P. KÃűnig, "Overt attention and context factors: The impact of repeated presentations, image type, and individual motivation," *PLOS ONE*, vol. 6, pp. 1–15, 07 2011. https://doi.org/10.1371/journal.pone.0021719.

[75] J. Rauthmann, C. T. Seubert, P. Sachse, and M. Furtner, "Eyes as windows to the soul: Gazing behavior is related to personality," *Journal of Research in Personality*, vol. 46, pp. 147–156, 04 2012. https://doi.org/10.1016/j.jrp.2011.12.010.

[76] I. Rigas, L. Friedman, and O. Komogortsev, "Study of an extensive set of eye movement features: Extraction methods and statistical analysis," *Journal of Eye Movement Research*, vol. 11, Mar. 2018. https://doi.org/10.16910/jemr.11.1.3.

[77] M. Kodappully, B. Srinivasan, and R. Srinivasan, "Towards predicting human error: Eye gaze analysis for identification of cognitive steps performed by control room operators," *Journal of Loss Prevention in the Process Industries*, vol. 42, pp. 35–46, 2016. https://doi.org/10.1016/j.jlp.2015.07.001.

[78] L. Wang, "Glissadic saccades: a possible measure of vigilance," *Ergonomics*, vol. 41, no. 5, pp. 721–732, 1998. `https://doi.org/10.1080/001401398186874`.

[79] A. T. Bahill and L. Stark, "Overlapping saccades and glissades are produced by fatigue in the saccadic eye movement system," *Experimental neurology*, vol. 48, no. 1, pp. 95–106, 1975. `https://doi.org/10.1016/0014-4886(75)90225-3`.

[80] F. A. Alsulaiman and A. El Saddik, "Three-dimensional password for more secure authentication," *IEEE Transactions on Instrumentation and Measurement*, vol. 57, pp. 1929–1938, Sep. 2008. `https://doi.org/10.1109/TIM.2008.919905`.

[81] F. Aloul, S. Zahidi, and W. El-Hajj, "Two factor authentication using mobile phones," in *2009 IEEE/ACS International Conference on Computer Systems and Applications*, pp. 641–644, IEEE, 2009. `https://doi.org/10.1109/AICCSA.2009.5069395`.

[82] A. T. B. Jin, D. N. C. Ling, and A. Goh, "Biohashing: two factor authentication featuring fingerprint data and tokenised random number," *Pattern recognition*, vol. 37, no. 11, pp. 2245–2255, 2004. `https://doi.org/10.1016/j.patcog.2004.04.011`.

[83] P. T. Schultz and R. A. Sartini, "Method and system for multi-factor biometric authentication based on different device capture modalities," Aug. 4 2015. US Patent 9,100,825. `https://patents.google.com/patent/US20130227651A1/en`.

[84] B. Spector, "Multi-factor biometric authentication," Mar. 8 2007. US Patent App. 11/217,074. `https://patents.google.com/patent/US20130227651A1/en`.

[85] K. Abhishek, S. Roshan, P. Kumar, and R. Ranjan, "A comprehensive study on multifactor authentication schemes," in *Advances in Computing and Information Technology*, pp. 561–568, Springer, 2013. `https://bit.ly/2FVWrXq`.

[86] J. Poplett, "User-friendly multifactor mobile authentication," Feb. 4 2014. US Patent 8,646,056. `https://patents.google.com/patent/US20080289030`.

[87] A. Ometov and S. Bezzateev, "Multi-factor authentication: A survey and challenges in v2x applications," in *2017 9th International Congress on Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT)*, pp. 129–136, IEEE, 2017. `https://doi.org/10.1109/ICUMT.2017.8255200`.

[88] A. Rosati, "Two factor authentication using near field communications," Mar. 14 2017. US Patent 9,594,896. `https://doi.org/10.1109/THS.2016.7568941`.

[89] N. Gunson, D. Marshall, H. Morton, and M. Jack, "User perceptions of security and usability of single-factor and two-factor authentication in automated telephone banking," *Computers & Security*, vol. 30, no. 4, pp. 208–220, 2011. `https://doi.org/10.1016/j.cose.2010.12.001`.

[90] J. Zhang, X. Tan, X. Wang, A. Yan, and Z. Qin, "T2fa: Transparent two-factor authentication," *IEEE Access*, vol. 6, pp. 32677–32686, 2018. `https://doi.org/10.1109/ACCESS.2018.2844548`.

[91] Y. Xiong and J. J. LaViola Jr, "A shortstraw-based algorithm for corner finding in sketch-based interfaces," *Computers & Graphics*, vol. 34, no. 5, pp. 513–527, 2010. `https://doi.org/10.1016/j.cag.2010.06.008`.

[92] D. H. Kim and M.-J. Kim, "A curvature estimation for pen input segmentation in sketch-based modeling," *Computer-Aided Design*, vol. 38, no. 3, pp. 238–248, 2006. `https://doi.org/10.1016/j.cad.2005.10.006`.

[93] P. I. Good, *Resampling methods*. Springer, 2006. `https://doi.org/10.1007/0-8176-4444-X`.

[94] U. Lall and A. Sharma, "A nearest neighbor bootstrap for resampling hydrologic time series," *Water Resources Research*, vol. 32, no. 3, pp. 679–693, 1996. `https://doi.org/10.1029/95WR02966`.

[95] N. Weber, "On resampling techniques for regression models," *Statistics & probability letters*, vol. 2, no. 5, pp. 275–278, 1984. https://doi.org/10.1016/0167-7152(84)90064-6.

[96] T. Hammond and B. Paulson, "Recognizing sketched multistroke primitives," *ACM Transactions on Interactive Intelligent Systems (TiiS)*, vol. 1, no. 1, p. 4, 2011. http://dx.doi.org/10.1145/2030365.2030369.

[97] B. Paulson and T. Hammond, "A system for recognizing and beautifying low-level sketch shapes using ndde and dcr," in *ACM Symposium on User Interface Software and Technology (UIST2007)*, 2007. https://bit.ly/2YyV0FH.

[98] T. Hammond and R. Davis, "Creating the perception-based ladder sketch recognition language," in *Proceedings of the 8th ACM Conference on Designing Interactive Systems*, pp. 141–150, ACM, 2010. https://dl.acm.org/citation.cfm?doid=1858171.1858197.

[99] T. A. Hammond and R. Davis, "Recognizing interspersed sketches quickly," in *Proceedings of Graphics Interface 2009*, pp. 157–166, Canadian Information Processing Society, 2009. https://doi.org/10.1145/1555880.1555917.

[100] T. Hammond and R. Davis, "Ladder: A language to describe drawing, display, and editing in sketch recognition," in *Proceedings of the International Joint Conference on Aritificial Intelligence (IJCAI)*, (Alcapulco, Mexico, Mexico), pp. 461–467, AAAI, 2003. https://doi.org/10.1145/1185657.1185788.

[101] T. Hammond and R. Davis, "Shady: A shape description debugger for use in sketch recognition," in *AAAI Fall Symposium on Making Pen-Based Interaction Intelligent and Natural*, p. 7, 2004. https://bit.ly/2NDTH7h.

[102] T. Hammond and R. Davis, "Automatically transforming symbolic shape descriptions for use in sketch recognition," in *AAAI*, vol. 4, pp. 450–456, 2004. https://www.aaai.org/Papers/AAAI/2004/AAAI04-072.pdf.

[103] T. Hammond and R. Davis, "Ladder, a sketching language for user interface developers," in *Computers & Graphics*, vol. 29-4, pp. 518–532, Elsevier, 2005. `https://dl.acm.org/ft_gateway.cfm?id=1281546&type=pdf`.

[104] R. Morris and K. Thompson, "Password security: A case history," *Communications of the ACM*, vol. 22, no. 11, pp. 594–597, 1979. `https://dl.acm.org/citation.cfm?id=359172`.

[105] T. Hammond and R. Davis, "Interactive learning of structural shape descriptions from automatically generated near-miss examples," in *Proceedings of the 11th international conference on Intelligent user interfaces*, pp. 210–217, ACM, 2006. `https://dl.acm.org/citation.cfm?id=1111495`.

[106] D. Rubine, "Specifying gestures by example," *SIGGRAPH Comput. Graph.*, vol. 25, pp. 329–337, July 1991. `https://dl.acm.org/citation.cfm?id=122753`.

[107] M. Nyström and K. Holmqvist, "An adaptive algorithm for fixation, saccade, and glissade detection in eyetracking data," *Behavior research methods*, vol. 42, no. 1, pp. 188–204, 2010. `https://doi.org/10.3758/BRM.42.1.188`.

[108] D. D. Salvucci and J. H. Goldberg, "Identifying fixations and saccades in eye-tracking protocols," in *Proceedings of the 2000 symposium on Eye tracking research & applications*, pp. 71–78, ACM, 2000. `https://dl.acm.org/citation.cfm?id=355028`.

[109] R. Andersson, L. Larsson, K. Holmqvist, M. Stridh, and M. Nyström, "One algorithm to rule them all? an evaluation and discussion of ten eye movement event-detection algorithms," *Behavior research methods*, vol. 49, no. 2, pp. 616–637, 2017. `https://doi.org/10.3758/s13428-016-0738-9`.

[110] I. Guyon and A. Elisseeff, "An introduction to variable and feature selection," *Journal of machine learning research*, vol. 3, no. Mar, pp. 1157–1182, 2003. `http://www.jmlr.org/papers/volume3/guyon03a/guyon03a.pdf`.

[111] R. Kohavi and G. H. John, "Wrappers for feature subset selection," *Artificial intelligence*, vol. 97, no. 1-2, pp. 273–324, 1997. `https://doi.org/10.1016/S0004-3702(97)00043-X`.

[112] G. Chandrashekar and F. Sahin, "A survey on feature selection methods," *Computers & Electrical Engineering*, vol. 40, no. 1, pp. 16–28, 2014. `https://doi.org/10.1016/j.compeleceng.2013.11.024`.

[113] Y. Yang and J. O. Pedersen, "A comparative study on feature selection in text categorization," in *Icml*, vol. 97, p. 35, 1997. `https://dl.acm.org/citation.cfm?id=657137`.

[114] Y. Benjamini and Y. Hochberg, "Controlling the false discovery rate: a practical and powerful approach to multiple testing," *Journal of the Royal statistical society: series B (Methodological)*, vol. 57, no. 1, pp. 289–300, 1995. `https://www.jstor.org/stable/2346101?seq=1#page_scan_tab_contents`.

[115] B. Thompson, "Factor analysis," *The Blackwell Encyclopedia of Sociology*, 2007. `https://doi.org/10.1002/9781405165518.wbeosf003`.

[116] J. R. Quinlan, "Induction of decision trees," *Machine learning*, vol. 1, no. 1, pp. 81–106, 1986. `https://link.springer.com/article/10.1007/BF00116251`.

[117] N. V. Cuong, V. Dinh, and L. S. T. Ho, "Mel-frequency cepstral coefficients for eye movement identification," in *2012 ieee 24th international conference on tools with artificial intelligence*, vol. 1, pp. 253–260, IEEE, 2012. `https://doi.org/10.1109/ICTAI.2012.42`.

[118] L. Fei-Fei, R. Fergus, and P. Perona, "One-shot learning of object categories," *IEEE transactions on pattern analysis and machine intelligence*, vol. 28, no. 4, pp. 594–611, 2006. `https://doi.org/10.1109/TPAMI.2006.79`.

[119] A. Santoro, S. Bartunov, M. Botvinick, D. Wierstra, and T. Lillicrap, "One-shot learning with memory-augmented neural networks," *arXiv preprint arXiv:1605.06065*, 2016. https://arxiv.org/abs/1605.06065.

[120] L. Fe-Fei *et al.*, "A bayesian approach to unsupervised one-shot learning of object categories," in *Proceedings Ninth IEEE International Conference on Computer Vision*, pp. 1134–1141, IEEE, 2003. https://dl.acm.org/citation.cfm?id=946776.