# COMPARING ACCURACY AND TIME COMPLEXITY OF MACHINE LEARNING ALGORITHMS FOR EYE GESTURE RECOGNITION

An Undergraduate Research Scholars Thesis

by

JIAYAO LI

Submitted to the Undergraduate Research Scholars program at
Texas A&M University
in partial fulfillment of the requirements for the designation as an

UNDERGRADUATE RESEARCH SCHOLAR

Approved by Research Advisor: Dr. Tracy Hammond

May 2018

Major: Computer Science and Engineering

# TABLE OF CONTENTS

Page

# ABSTRACT

Comparing Accuracy and Time Complexity of Machine Learning Algorithms for Eye Gesture Recognition

Jiayao Li
Department of Computer Science and Engineering
Texas A&M University

Research Advisor: Dr. Tracy Hammond
Department of Computer Science and Engineering
Texas A&M University

The eye motion data can be utilized to perform behavior analysis and improve common applications, such as accessible HCI, interactive interface, marketing, and remote-controlling. This research project compares accuracy and time complexity of three commonly used machine learning algorithms for eye gesture recognition. The importance of this project is to examine ways to improve efficiency in recognizing eye gestures. It was found that the template matching algorithm has the best accuracy, followed by the Pearson correlation algorithm, and lastly the decision tree algorithm. For time performance, it was found that the decision tree algorithm performs the best, closely followed by the Pearson correlation algorithm, and lastly the template matching algorithm. The template matching algorithm is recommended to be used in accuracy-sensitive situations. The decision tree algorithm and the Pearson correlation algorithm are recommended for time-sensitive situations. The algorithms perform better when the directions and other relative properties of input gestures are majorly different. One should consider the properties of the input gesture and the nature of application when it comes to deciding which algorithm to use.

# DEDICATION

I would like to dedicate this paper to my parents, who have always supported me. I love both of you dearly.

# ACKNOWLEDGMENTS

First and foremost, I would like to express my gratitude to my faculty advisor, Dr. Tracy Hammond, for her unwavering support, professional guidance, and excellent teaching throughout this process. I would also like to thank my graduate mentor, Vijay Rajanna. This research would not have been possible without Vijay's continuous support and guidance. I would also like to give thanks to Paul Taele for kindly helping me with the thesis formatting and the reviewing process. Last but not least, I would like to give thanks to the members of the Sketch Recognition Lab and all of the user study participants.

# NOMENCLATURE

| | |
|---|---|
| ACC | Accuracy |
| DT | Decision Tree |
| HCI | Human-Computer Interaction |
| PC | Pearson Correlation |
| TM | Template Matching |

# LIST OF FIGURES

# LIST OF TABLES

TABLE Page

# CHAPTER 1

# INTRODUCTION

Human eyes move in particular motions and the movement corresponds with the attention of interest. The brain generates stimuli from the vision and processes sensory input by concentrating on focal points [1]. The ability to utilize gaze data effectively can lead to highly rich interactions. Gaze input has great potential as it moves faster than a mouse [2] and can be used to replace the mouse for tasks involving object selection [3, 4, 5]. Eye-tracking is simply tracing the path of a person's eye movements. The equipment can be built into the computer monitor or be placed in front of the computer screen. While the user is looking at the screen, the eye-tracking software gathers the user's focal points data on the screen. The technology has many useful applications. Common applications include desktop control [6, 7, 8, 9], typing [10, 11, 12, 13, 14], target selection [15, 16, 17], entering password [18, 19], game control [20, 21], task prediction [22], visual analytics [23], and giving commands at a distance [24].

Eye-tracking has a wide range of application areas. It can be used as a direct replacement for the mouse [4] and is particularly suitable for applications with limited use of keyboard input but rely greatly on the mouse input [5]. Eye-tracking also has promising potential to be used as an interaction method with remote displays [24]. Gaze-gesture based interaction system has several advantages than the traditional dwell-time based interaction system. Gaze-gesture based interaction system can achieve high accuracy even with disturbed calibration and does not require the users to constantly repeat the calibration process [25]. This paper uses gaze-gesture as an input method to evaluate the accuracy performance and time complexity of three different machine learning algorithms. Studies have been done on performance comparison among classification algorithms [26] and routing algorithms [27]. However, there has not been much research done on performance comparison of the commonly used eye-tracking algorithms for gaze-recognition.

There are several commonly used algorithms when it comes to gaze-recognition. This research project compares the accuracy and time complexity of three commonly used machine learning algorithms for eye-gesture recognition: the template matching algorithm [28], the decision tree algorithm [29], and the Pearson correlation algorithm [30]. Six distinctive gestures were implemented as the base gesture types. These gestures were used as training templates for the machine learning algorithms. A tabletop eye-tracker from "The Eye Tribe" was used to collect the user's gaze data. The eye-tracker was placed directly in front of the computer screen. Upon the activation of the user's command by pressing down a key, the computer screen started to plot the user's eye movement data until the key is released by the user. The users were instructed to use their eyes to draw each of the six base gesture types. There were 22 users participated in the study. The user's gaze gesture was then classified by each of the three machine learning algorithms. The classification result and the time measurement of each algorithm were recorded for data analysis and result comparison.

This paper has the following contributions. The accuracy performance and time complexity of the three machine learning algorithms were determined. For each algorithm, a confusion matrix was produced and the F-measure was calculated for each of the input gesture types. The effects of input gestures and base gestures were discussed. Recommendations on algorithm selection were made based on the algorithm performance observed in the study. Similar analyses can be conducted on related applications to achieve computing efficiency.

# CHAPTER 2

# RELATED WORK

Eye-tracking has been used in Human-Computer Interaction (HCI) for both accessible and rich interactions. There are many useful applications when it comes to eye-tracking. This section covers some of the major applications of eye-tracking in HCI.

## 2.1 Accessible HCI

Previous research has used gaze input as an accessible HCI, allowing users with accessibility needs to use their gaze gesture to perform computer actions [31]. Users with motor impairments can use gaze gestures to enter text on a computer using a virtual keyboard with the assistance of an eye-tracker [11, 32]. Gaze typing allows individuals with motor impairments to enter characters by using the duration of the dwell time on a virtual keyboard as the user dwells on a specific key [11]. Gaze-based typing system has also been advanced by implementing a wearable foot operated device, where the user can select a character with the assistance of wearable technology [32].

## 2.2 Authentication

Gaze gesture input is also commonly used for authentication [33, 34, 35, 36]. The gaze input can be used as a powerful security tool to prevent shoulder-surfing attacks [25]. Shoulder surfing allows an attacker to access the authentication information through observation and has become a threat to visual privacy [33]. A gaze-based user authentication system that combines gaze with gesture recognition can effectively prevent shoulder surfing attacks [25]. The eye gesture allows for real-time user authentication without the need of physically entering passwords through a keypad, effectively improving the security of the authentication process.

## 2.3 Engineering

Eye-tracking also has a lot of useful applications in engineering such as controlling an airplane or conducting an inspection [1, 37, 38]. Eye-tracking allows the users to input gestures at a distance and carry out field tasks more conveniently [39, 40, 41]. Eye-tracking can be used in situations where the interface involves sophisticated control panels or when remote controlling is needed, such as when the control is too hot to touch or when it is hard to reach. Eye-tracking provides flexibility in engineering design and allows engineering systems to be more efficient.

## 2.4 Large Screen Interaction

Eye gesture input is also commonly used for large screen interaction [42, 41]. The gaze input method allows the users to be more engaging [43]. Public displays such as a museum interface, poster sign, map board [44, 45, 46] can all utilize input gesture as a trigger to initiate interaction. Gaze-input allows the interactive system to engage with large audience efficiently. The users can use their gaze to interact with the interface without having to have their own input device. Utilizing gaze gesture can effectively save time, cost, and bring more convenience.

## 2.5 Marketing

Gaze gesture has also been used as a powerful marketing analysis tool to collect data about user's level of interest on a web page or an object [47, 48, 49, 50, 23]. The inputs are particularly useful because it provides information about whether or not the intended design features were scanned over by the user. With the data provided by eye-tracking, one can measure the relationship between marketing actions and sales of product [37]. More effective marketing strategies and advertising methods can be developed with the utilization of gaze input.

## 2.6 Surgery

Gaze input can be used to assist in performing surgeries, where the hygiene need is critical and a surgeon may be busy with other tasks with their hands [51, 52]. Eye-tracking technology brings great potential for touches interaction techniques in medical settings [51]. Eye-tracking can also be utilized as a tool for assessing surgical skill [53] or as a potential training tool [54] in clinical surgery. A previous study suggests that the tool-motion data and the eye-gaze data can be used to effectively evaluate a surgeon's surgical skill [53]. In a training environment, eye-tracking can be used as an effective tool to provide a supervisor's eye-gaze data as a visual instruction to the trainee [54].

## 2.7 Video Game Control

Eye-tracking can also be used as an input method for video games. The gaze input data can not only provide information about the user's points of focus, but can also be used to estimate the user's head orientation [21]. A previous study suggests that using eye-tracking can increase the immersion of a video game and improve the gaming experience [21].

## 2.8 Visual Analytics

Eye-tracking can also be used to collect gaze input as a form of feedback for visual analytics [23]. Artists value the feedback of what parts of their artwork is most appreciated by the viewers. Traditionally, this information is collected from the viewers in the form of oral or written feedback. However, the value of this feedback can be limited due to the lack of participation from the viewers, additionally, our subconscious visual understanding can sometimes be difficult to express verbally [23]. With the eye-tracking technology, artists can receive feedback in the form of visualized gaze input data that indicates areas of interest by the viewers [23].

# CHAPTER 3

# METHODS

There are various advantages of using gaze gestures than just using dwell time. Gaze-gesture based interaction system can achieve high accuracy even with disturbed calibration [25], which means that the gaze input does not need to be precise and the users do not have to constantly repeat the calibration process in order to effectively interact with the system.

## 3.1 Input Gestures

This research compares the time complexity of three different machine learning algorithms: the template matching algorithm, the decision tree algorithm, and the Pearson correlation algorithm. Six gestures were implemented as the base gesture types, shown in Figure 3.1.

These gestures were chosen because they are distinctively different from each other. Each gesture was trained with five training templates collected from five different individuals. An eye-tracker was used to plot the eye-movement data points against the computer screen. The algorithms were written in the C# language using Visual Studio. The operating system used was Microsoft Windows 10.

## 3.2 Template Matching Algorithm

The first algorithm is the template matching algorithm. This algorithm compares the user's gesture against the existing templates and determines the user's gesture by finding the best match [28]. The template matching algorithm compares two paths and calculates the distance of the input path from the template path [33]. When the input path matches

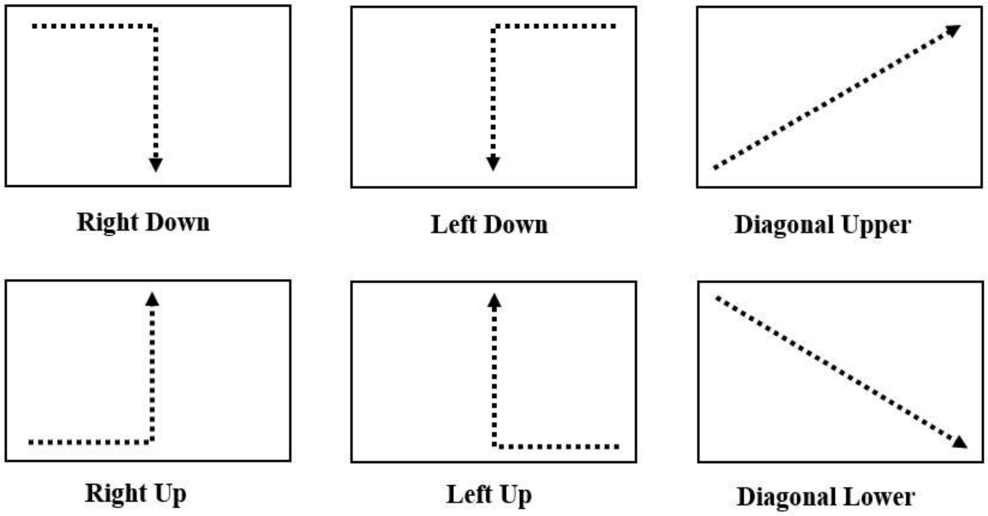


Figure 3.1: Six gesture types used as training templates.

closely to a template path, the gesture type of the template path is recognized as the input gesture type.

The first step of the algorithm is the sampling stage, where the offset and the total length between two points on a path are computed [33]. The purpose of the sampling stage is for all the input strokes to have an equal number of stroke points. Figure 3.2 shows the user's scan path being scaled down to N=64 points in the sampling stage [25].

The next step is the scaling stage, where the path is scaled to a square along the *x* and the *y* axes. Then, the centroid of the path is located and the path is moved to the origin point [33]. Once the input gesture path is processed, it is ready to be compared to all template gestures. The following equation is used for comparison [55]:

$$D = \sum_{i=1}^{n} \frac{\sqrt{(Input(i)_x - Template(i)_x)^2 + (Input(i)_y - Template(i)_y)^2}}{n} \tag{3.1}$$

where *D* stands for the distance between the input path and the template path, *i* is the point on the path, and *n* is the total sample size.

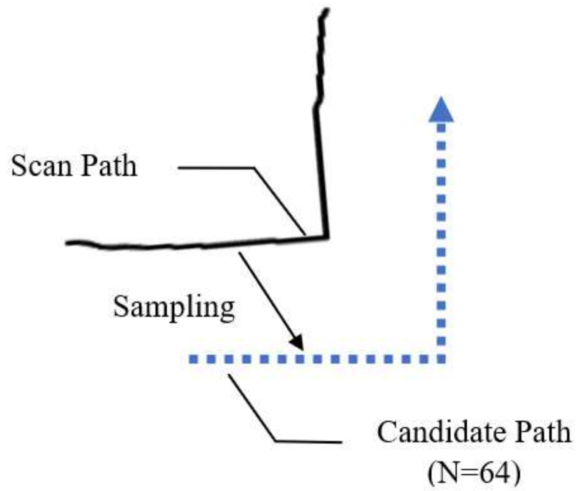


Figure 3.2: Demonstration of the scan path being scaled down to N=64 points.

Figure 3.3 shows a demonstration of the template matching algorithm finding the Euclidean distance between the candidate path and the template path [25].

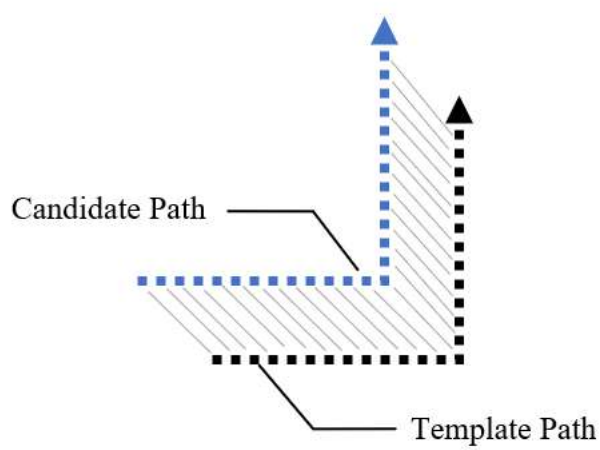


Figure 3.3: Demonstration of the candidate path being matched to the template path.

### 3.3 Decision Tree Algorithm

The second algorithm is the decision tree algorithm. This algorithm recognizes user's gesture by computing a range of features from the user's data points and comparing them with the computed features of the templates [29]. The decision tree algorithm used in this research analyzes a total of five gesture features: the start and the end point of a gesture, the area of the bounding box, the length of the bounding box diagonal, and the slope of the bounding box diagonal [55]. Figure 3.4 shows a demonstration of the computed features.

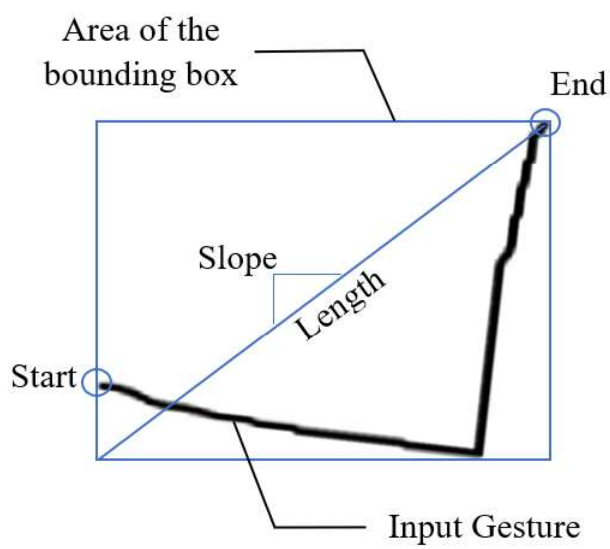


Figure 3.4: Demonstration of the accounted features in decision tree.

### 3.4 Pearson Correlation Algorithm

The third algorithm is the Pearson correlation algorithm. This algorithm measures the linear correlation between the user's input and the existing templates, shown in the equation below [30]:

$$C = \frac{Cov(Eye, Obj)}{\sigma_{Eye} \cdot \sigma_{Obj}} = \frac{\sum_{i=1}^{n}(Eye_i - \mu_{Eye})(Obj_i - \mu_{Obj})}{\sqrt{\sum_{i=1}^{n}(Eye_i - \mu_{Eye})^2}\sqrt{\sum_{i=1}^{n}(Obj_i - \mu_{Obj})^2}} \tag{3.2}$$

where *C* stands for the correlation coefficient, *Eye* denotes the data points of the user's eye-movements, *Obj* denotes the data points of the template object, *Cov(Eye,Obj)* is the covariance, $\sigma_{Eye}$ and $\sigma_{Obj}$ are the standard deviations, $Eye_i$ and $Obj_i$ are the single samples indexed with *i*, $\mu_{Eye}$ and $\mu_{Obj}$ are the means of the sample sums for $Eye_i$ and $Obj_i$, respectively, and *n* is the sample size.

The same equation is used to calculate the correlation coefficients in both the *x* and the *y* axes. The total coefficient is calculated by adding the coefficient obtained from the *x* direction to the coefficient obtained from the *y* direction. For each template in the set of the existing training templates, a coefficient is calculated, and the template gesture type with the highest coefficient is determined to be the input gesture type.

### 3.5 Time Measurement

The execution time is measured using the *Stopwatch* property. The unit of measurement is in "ticks". According to Microsoft documentation [56], a tick is the smallest unit the *Stopwatch* timer can measure. A tick can be converted to seconds by using the *Frequency* field, which represents the number of ticks per second [57]. The field frequency is dependent on the installed hardware and the operating system [57]. In this study, a tick is used as the time measurement unit for the purpose of performance comparison.

### 3.6 User Study

After the algorithm implementation stage, user studies were conducted to test the accuracy and the time complexity of the three algorithms. A total number of 22 users participated in this study, aged from 18 to 30 with an average age of 22. There were 7 female users and 15 male users. There was 1 user who wore glasses. During the user study, an eye-tracker was placed in front of the computer monitor at the bottom of the computer screen. The users were asked to use their eyes to draw the six base gesture types. A key was pressed by the user to initiate the gaze gesture, and gaze data was plotted against the computer screen until the key was released by the user. The eye-tracker was calibrated each time before use. Figure 3.5 shows a demonstration of the front and side view of the user study set up.

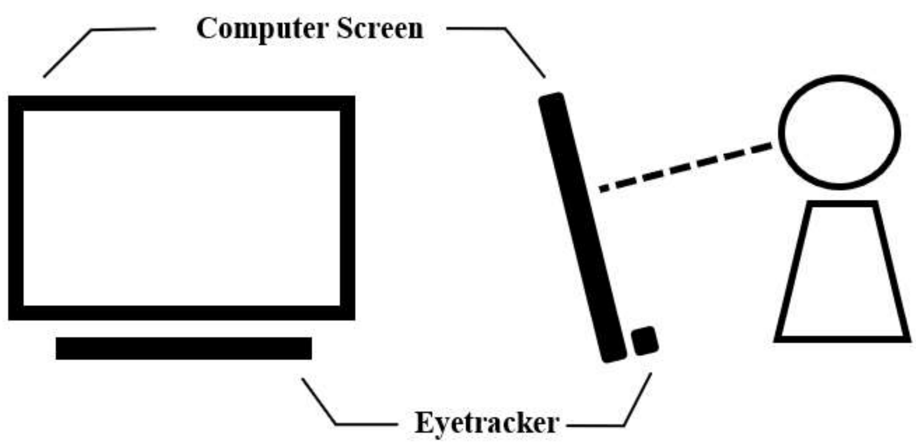


Figure 3.5: Demonstration of the front and side view of user study setup.

## 3.7 User Interface

Figure 3.6 shows a demonstration of the user interface. After a user performs a gesture on the screen, a message window would pop up upon command showing the classification and timing results by the three algorithms. As shown in the figure, a message window is displayed, indicating all three algorithms have successfully recognized the input gesture.

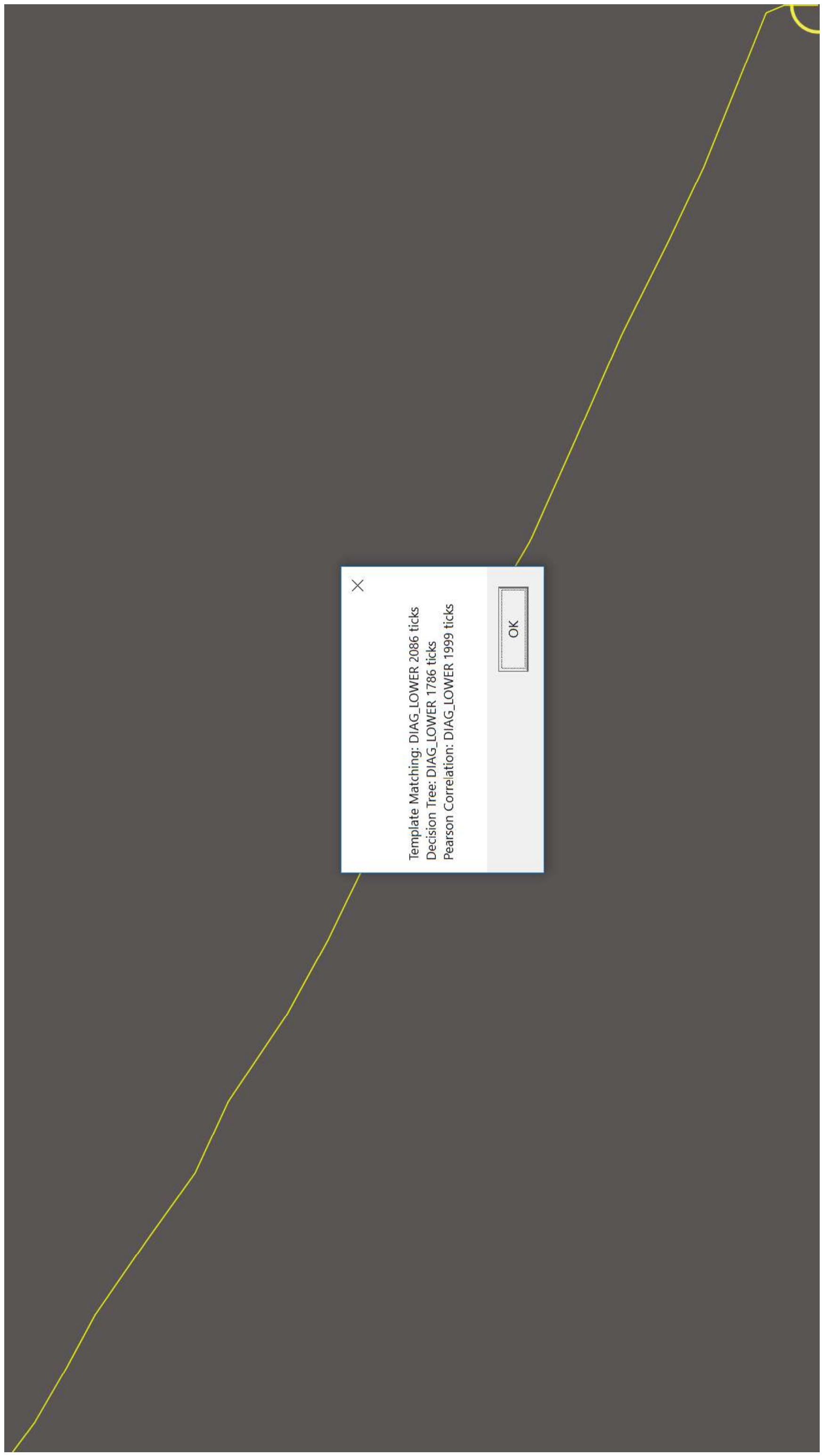


Figure 3.6: Demonstration of user interface.

# CHAPTER 4

# RESULTS

## 4.1 Gesture Type 1: Right-Down

Figure 4.1 shows the accuracy plot by the right-down gesture type. For this gesture type, both template matching and Pearson correlation achieve the most accuracy at around 91%, followed by decision tree at around 82%.

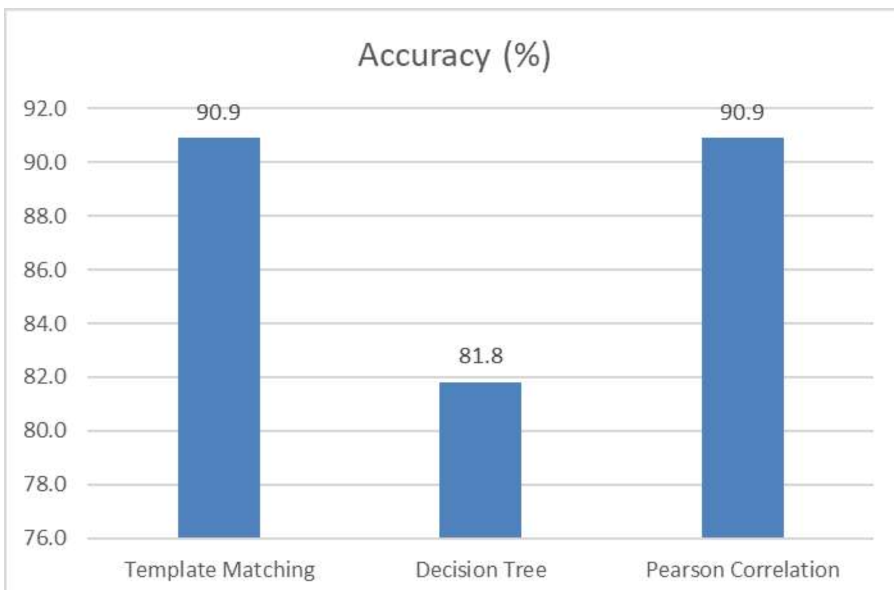


Figure 4.1: Accuracy plot by the right-down gesture type.

Figure 4.2 shows the average time plot by the right-down gesture type. For this gesture type, template matching takes the most time and Decision tree takes the least amount of time. Both decision tree and Pearson correlation perform significantly faster than template matching.

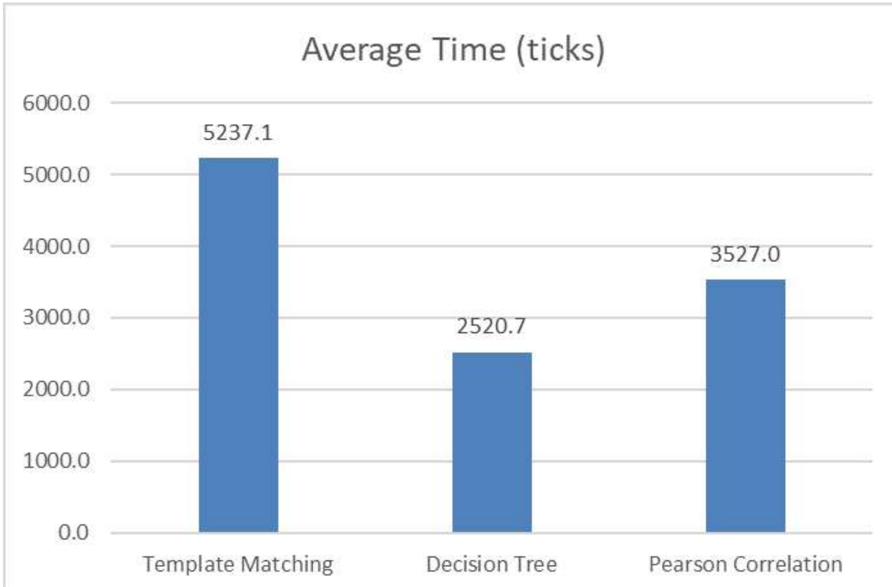


Figure 4.2: Average time plot by the right-down gesture type.

Figure 4.3 shows the scattered plot of the time data by the right-down gesture type. As shown in the plot, the template matching data points scatter mostly on the top of the graph, the Pearson correlation data points scatter mostly in the middle, and the decision tree data points scatter mostly at the bottom.

Table 4.1 shows a summary of the result for the right-down gesture type. For this gesture type, template matching achieves decent accuracy but consumes the most time, Pearson correlation achieves as much accuracy as template matching but performs much faster, and decision tree has the lowest accuracy but has the best time efficiency.

### 4.2 Gesture Type 2: Right-Up

Figure 4.4 shows the accuracy plot by the right-up gesture type. For this gesture type, template matching achieves the best accuracy at around 91%, followed by decision tree at

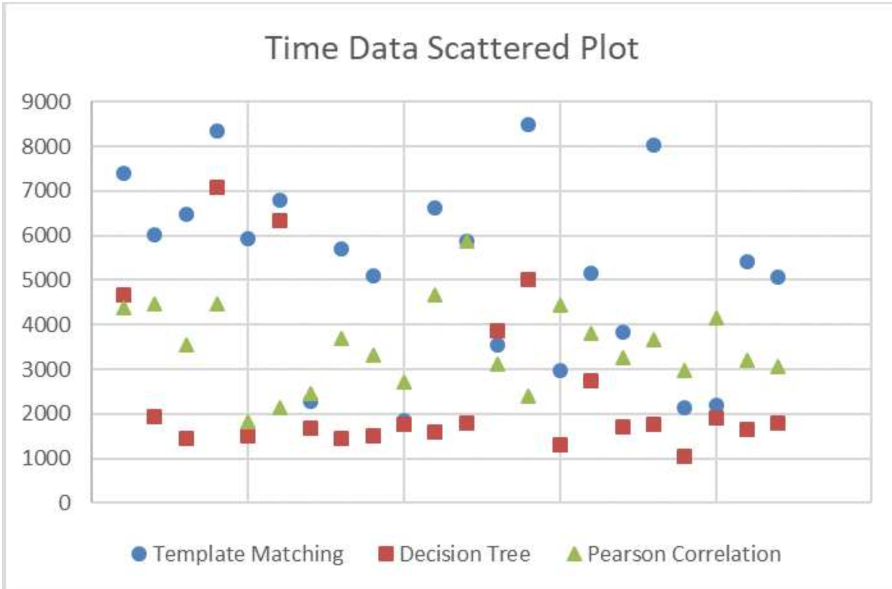


Figure 4.3: Scattered plot of time data by the right-down gesture type.

Table 4.1: Result summary of the right-down gesture type.

| | Template Matching | Decision Tree | Pearson Correlation |
|---|---|---|---|
| Accuracy (%) | 90.9 | 81.8 | 90.9 |
| Average Time (ticks) | 5,237.1 | 2,520.7 | 3,527.0 |

around 86%, and lastly Pearson correlation at around 73%.

Figure 4.5 shows the average time plot by the right-up gesture type. For this gesture type, template matching takes the most time, and Pearson correlation takes the second longest while decision tree takes the least amount of time. Both decision tree and Pearson correlation are found to be significantly faster than template matching.

Figure 4.6 shows the scattered plot of the time data by the right-up gesture type. As shown in the plot, the template matching data points scatter mostly on the top of the graph, the Pearson correlation data points scatter mostly in the middle, while the decision tree data points scatter mostly at the bottom.

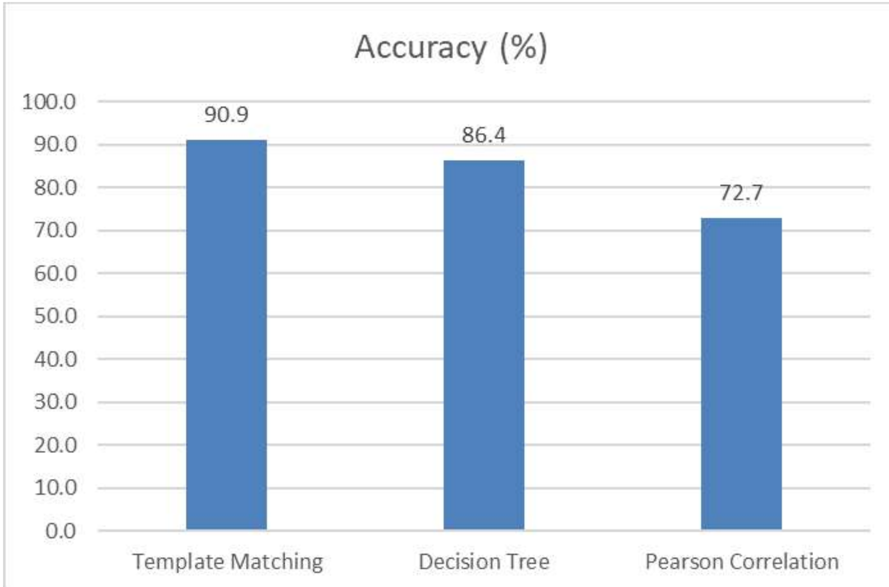


Figure 4.4: Accuracy plot by the right-up gesture type.

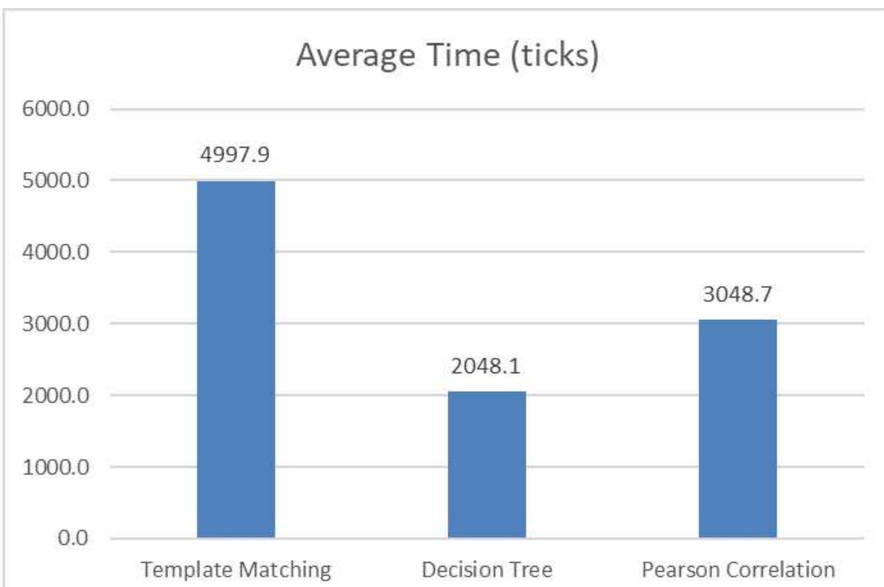


Figure 4.5: Average time plot by the right-up gesture type.

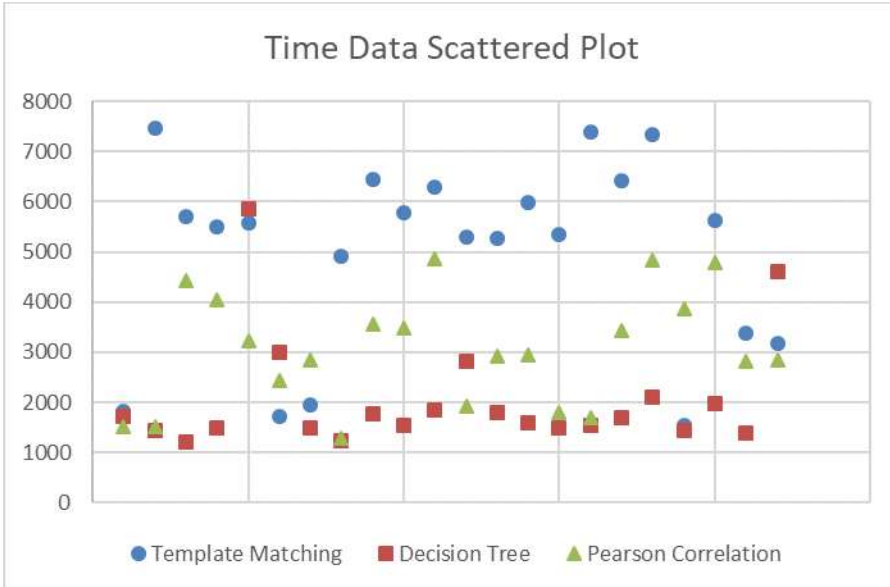


Figure 4.6: Scattered plot of time data by the right-up gesture type.

Table 4.2 shows a summary of the result for the right-up gesture type. For this gesture type, template matching achieves the best accuracy but takes the most time. Decision tree has a lower accuracy, not much lower than template matching, but has a much better time efficiency. Pearson correlation achieves the lowest accuracy but has a better time performance than template matching.

Table 4.2: Result summary of the right-up gesture type.

| | Template Matching | Decision Tree | Pearson Correlation |
|---|---|---|---|
| Accuracy (%) | 90.9 | 86.4 | 72.7 |
| Average Time (ticks) | 4,997.9 | 2,048.1 | 3,048.7 |

## 4.3 Gesture Type 3: Left-Down

Figure 4.7 shows the accuracy plot by the left-down gesture type. For this gesture type, all three algorithms achieve 100% of accuracy.

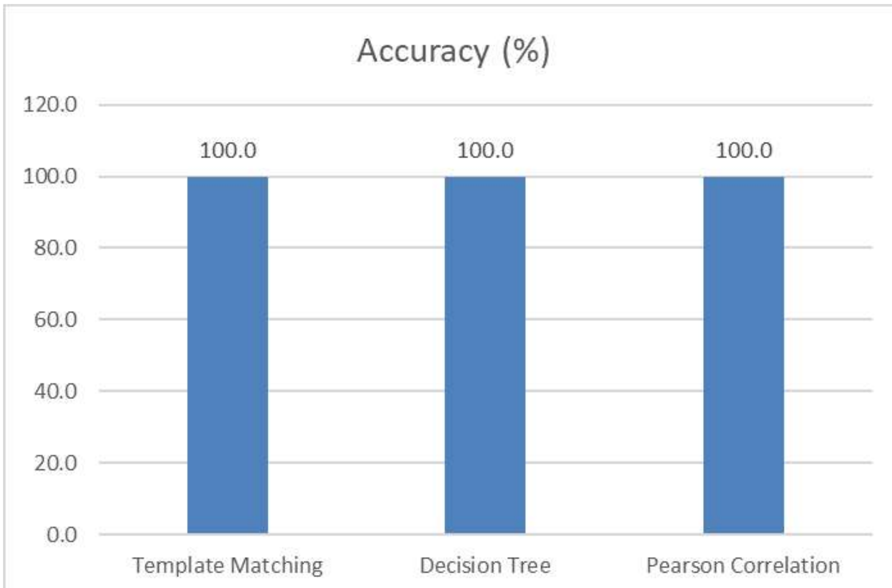


Figure 4.7: Accuracy plot by the left-down gesture type.

Figure 4.8 shows the average time plot by the left-down gesture type. For this gesture type, template matching takes the most time, decision tree takes the second most time, and Pearson correlation takes the least amount of time. The time performance of decision tree and Pearson correlation are close to each other, but both are significantly faster than template matching.

Figure 4.9 shows the scattered plot of the time data by the left-down gesture type. As shown in the plot, the template matching data points mostly occupy the upper half of the graph, while the data points of decision tree and Pearson correlation mostly occupy the lower half of the graph.

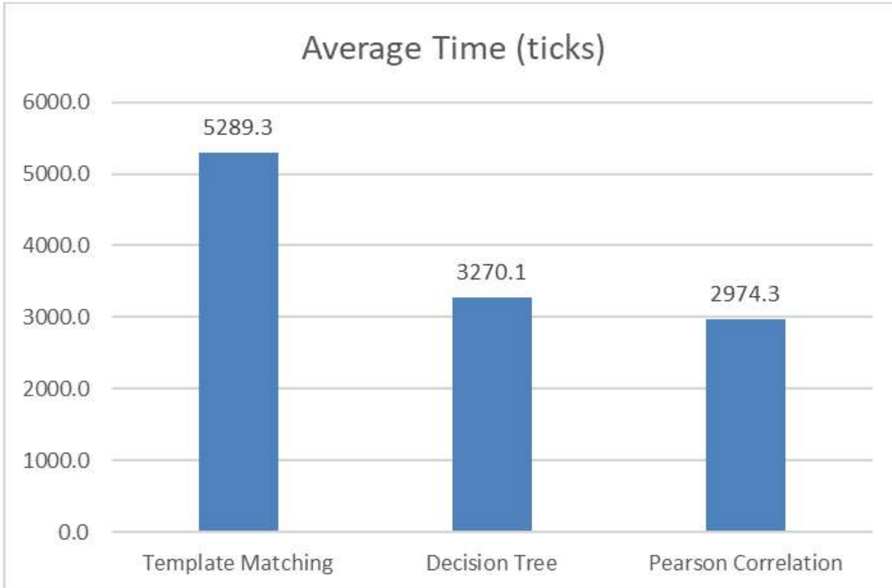


Figure 4.8: Average time plot by the left-down gesture type.

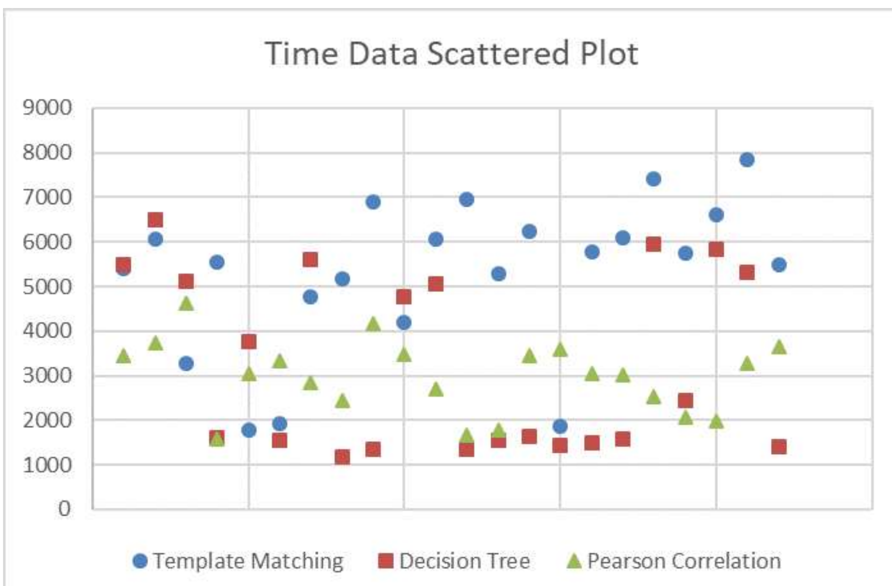


Figure 4.9: Scattered plot of time data by the left-down gesture type.

Table 4.3 shows a summary of the result for the left-down gesture type. For this gesture type, all three algorithms achieve 100% of accuracy. Both decision tree and Pearson correlation perform much better on time than template matching.

Table 4.3: Result summary of the left-down gesture type.

| | Template Matching | Decision Tree | Pearson Correlation |
|---|---|---|---|
| Accuracy (%) | 100 | 100 | 100 |
| Average Time (ticks) | 5,289.3 | 3,270.1 | 2,974.3 |

## 4.4 Gesture Type 4: Left-Up

Figure 4.10 shows the accuracy plot by the left-up gesture type. For this gesture type, both decision tree and Pearson correlation achieve 100% of accuracy, closely followed by template matching at around 96%.

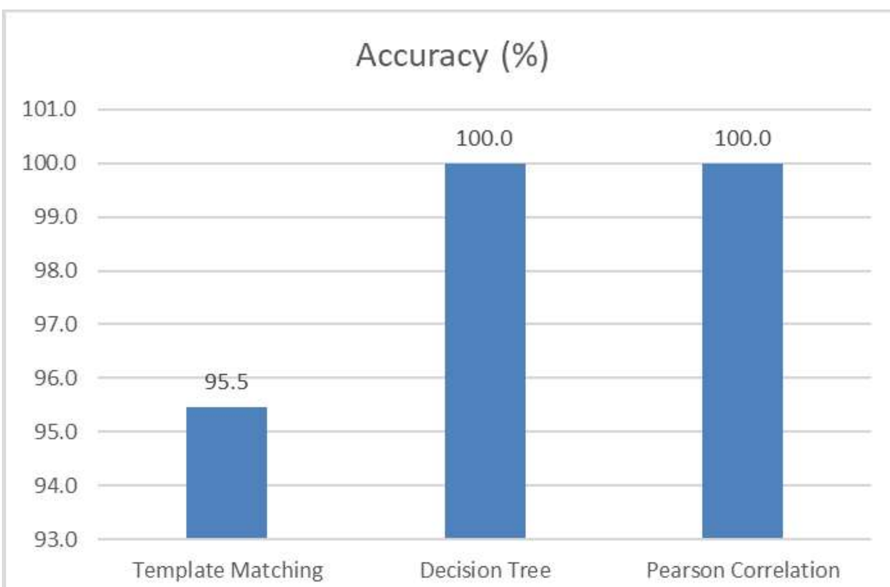


Figure 4.10: Accuracy plot by the left-up gesture type.

Figure 4.11 shows the average time plot by the left-up gesture type. For this gesture type, template matching takes the most time, and decision tree and Pearson correlation are both significantly faster than template matching. Decision tree takes the least amount of time, but not much lower than Pearson correlation.

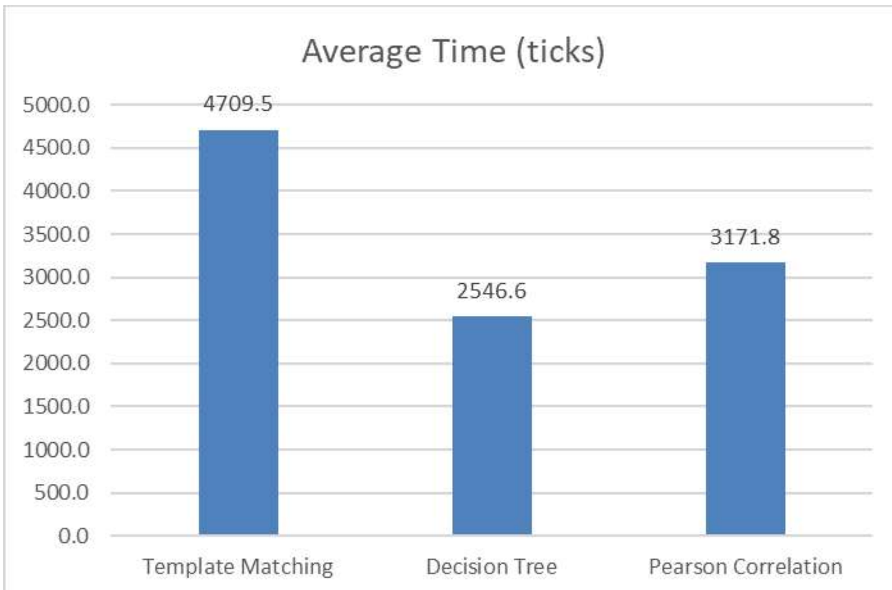


Figure 4.11: Average time plot by the left-up gesture type.

Figure 4.12 shows the scattered plot of the time data by the left-up gesture type. As shown in the plot, the template matching data points are found to be at a higher range than the other two algorithms. Decision tree has a few data points at the top but scatters mostly at the bottom.

Table 4.4 shows a summary of the result for the left-up gesture type. For this gesture type, all three algorithms achieve 100% of accuracy. Template matching takes the most time. Both decision tree and Pearson correlation are found to be faster than template matching%.

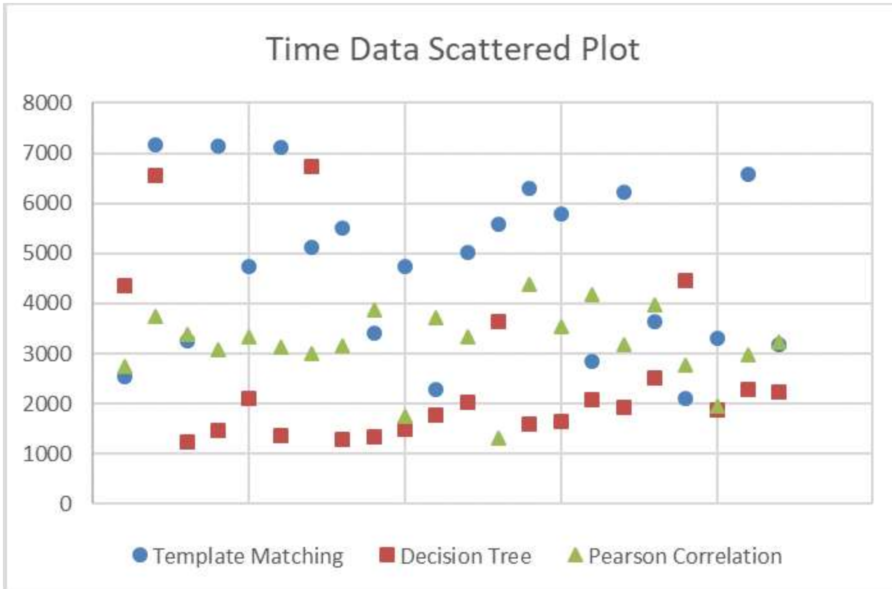


Figure 4.12: Scattered plot of time data by the left-up gesture type.

Table 4.4: Result summary of the left-up gesture type.

| | Template Matching | Decision Tree | Pearson Correlation |
|---|---|---|---|
| Accuracy (%) | 100 | 100 | 100 |
| Average Time (ticks) | 5,289.3 | 3,270.1 | 3,171.8 |

### 4.5 Gesture Type 5: Diagonal-Upper

Figure 4.13 shows the accuracy plot by the diagonal-upper gesture type. For this gesture type, Both template matching and Pearson correlation achieve 100% of accuracy. Decision tree only has an accuracy of about 86%.

Figure 4.14 shows the average time plot by the diagonal-upper gesture type. For this gesture type, decision tree has the best time performance, followed by Pearson correlation, and template matching is once again found to take the longest time.

Figure 4.15 shows the scattered plot of the time data by the diagonal-upper gesture type. As shown in the plot, the template matching data points scatter mostly on the upper half of

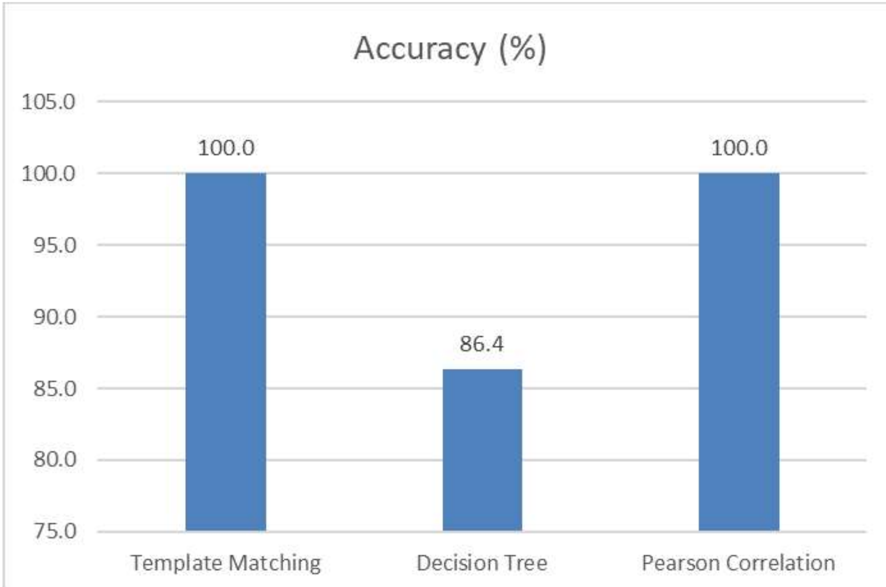


Figure 4.13: Accuracy plot by the diagonal-upper gesture type.

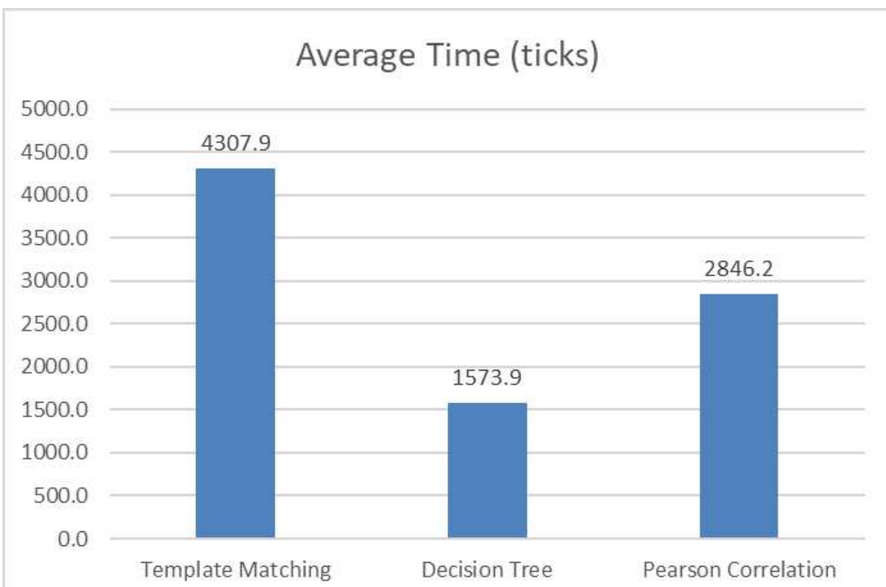


Figure 4.14: Average time plot by the diagonal-upper gesture type.

the graph while the decision tree data and the Pearson correlation data scatter mostly at the lower half of the graph.

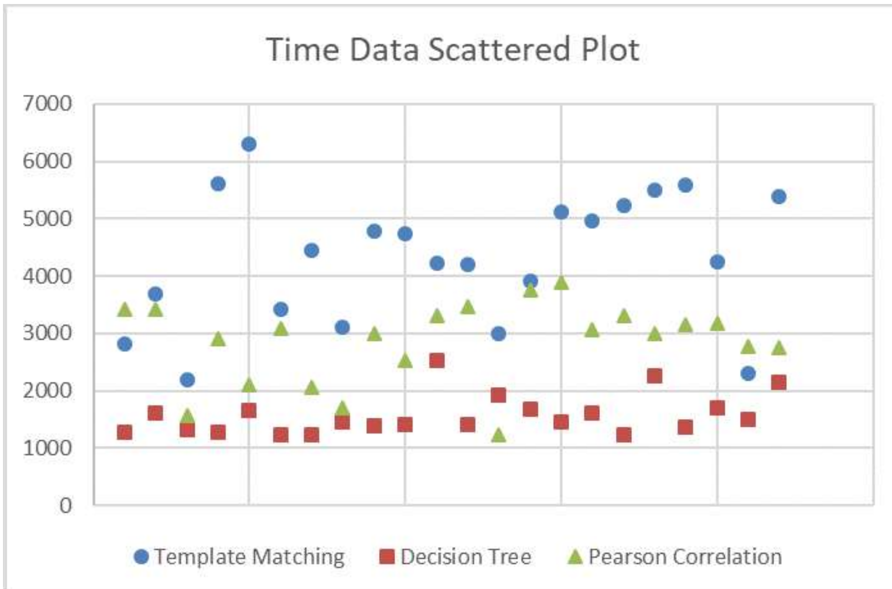


Figure 4.15: Scattered plot of time data by the diagonal-upper gesture type.

Table 4.5 shows a summary of the result for the diagonal-upper gesture type. For this gesture type, both template matching and Pearson correlation achieve 100% of accuracy. Decision tree takes the lowest amount of time but also has the lowest accuracy. Pearson correlation performs slightly slower than decision tree and much faster than template matching.

Table 4.5: Result summary of the diagonal-upper gesture type.

| | Template Matching | Decision Tree | Pearson Correlation |
|---|---|---|---|
| Accuracy (%) | 100 | 86.4 | 100 |
| Average Time (ticks) | 4,307.9 | 1,573.9 | 2,846.2 |

## 4.6 Gesture Type 6: Diagonal-Lower

Figure 4.16 shows the accuracy plot by the diagonal-lower gesture type. Once again, both template matching and Pearson correlation achieve the most accuracy at 100%. Decision tree has the lowest accuracy at around 86%.

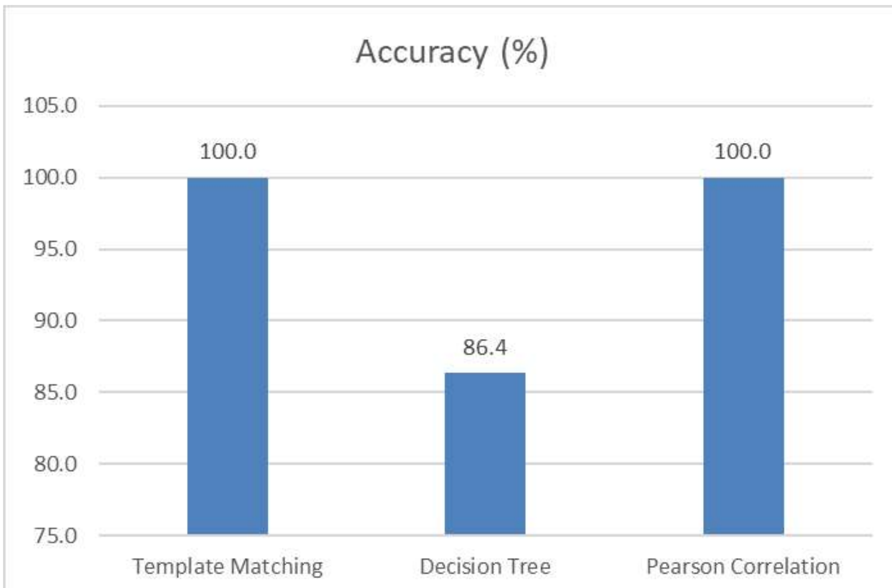


Figure 4.16: Accuracy plot by the diagonal-lower gesture type.

Figure 4.17 shows the average time plot by the diagonal-lower gesture type. For this gesture type, template matching takes the most time. Decision tree and Pearson correlation have almost the same time performance, both are much faster than template matching.

Figure 4.18 shows the scattered plot of the time data by the diagonal-lower gesture type. As shown in the plot, the template matching data points scatter at a higher range than the data points of decision tree and Pearson correlation.

Table 4.6 shows a summary of the result for the diagonal-lower gesture type. For this gesture type, both template matching and Pearson correlation achieve 100% of accuracy,

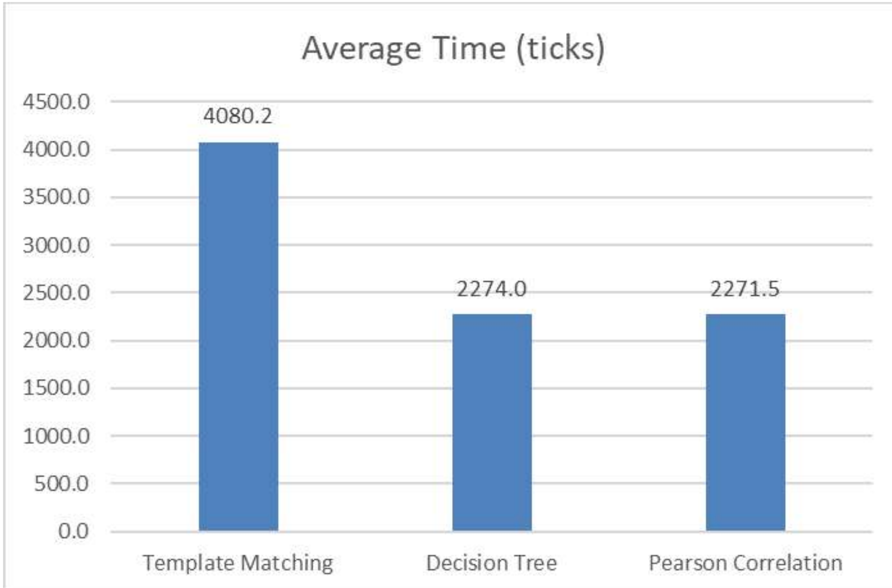


Figure 4.17: Average time plot by the diagonal-lower gesture type.

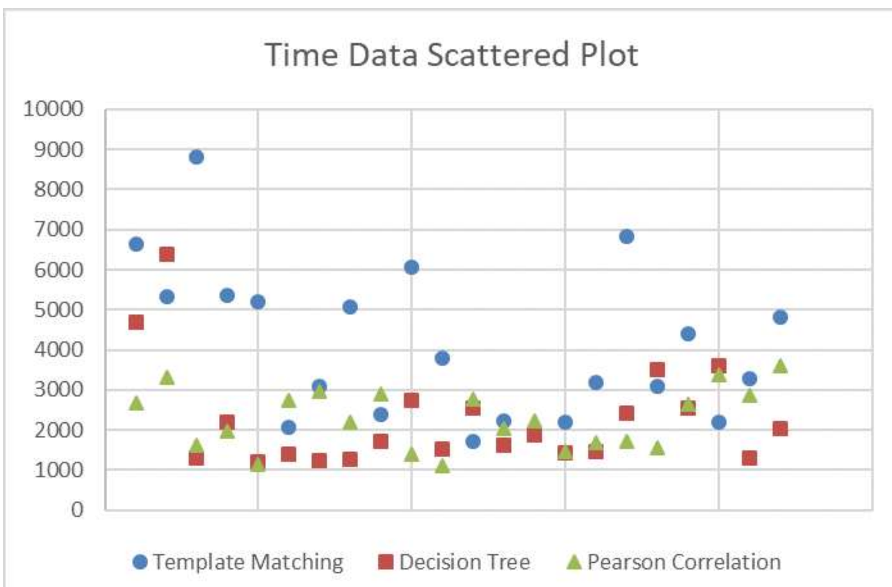


Figure 4.18: Scattered plot of time data by the diagonal-lower gesture type.

but template matching takes the most time. Decision tree has a good time performance but its accuracy is much lower than the other two algorithms.

Table 4.6: Result summary of the diagonal-lower gesture type.

| | Template Matching | Decision Tree | Pearson Correlation |
|---|---|---|---|
| Accuracy (%) | 100 | 86.4 | 100 |
| Average Time (ticks) | 4,080.2 | 2,274.0 | 2,271.5 |

### 4.7 All Gestures Combined

Figure 4.19 shows the accuracy plot by the all the gestures combined. As shown in the figure, when considering all the gestures, template matching achieves the best accuracy at around 96%, followed by Pearson correlation at around 94%, and lastly decision tree at around 90%.

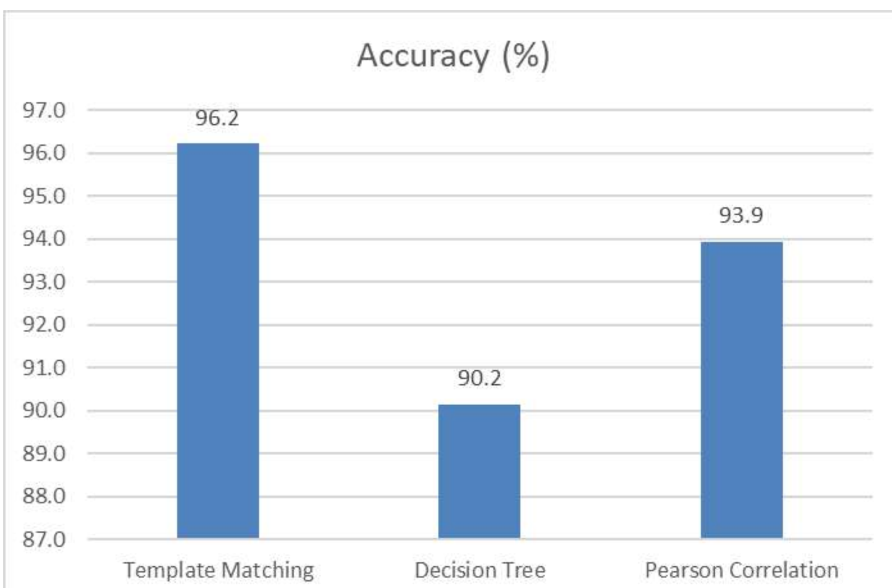


Figure 4.19: Accuracy plot by all gestures combined.

Figure 4.20 shows the average time plot by all the gestures combined. As shown in the graph, template matching takes the most amount of time. Both decision tree and Pearson correlation perform significantly better than template matching on time performance. Decision tree has the best time performance among all three algorithms.

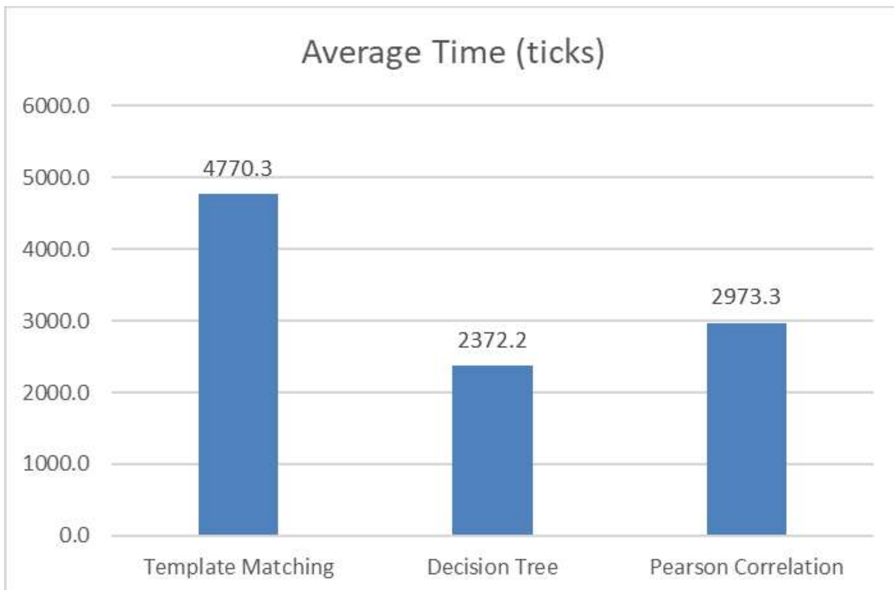


Figure 4.20: Average time plot by all gestures combined.

Figure 4.21 shows the scattered plot of the time data by all the gesture types combined. As shown in the plot, the template matching data points fall frequently on the top of the plot while the Pearson data points scatter mostly at the bottom. Decision tree has some data points falling on the upper half of the graph, but for the most part, the data points scatter at the bottom of the graph.

Table 4.7 shows a summary of the result for all the gesture types combined. When considering all the gesture types, template matching has the best performance but takes the most time. Decision tree takes the least amount of time but has the lowest accuracy. The

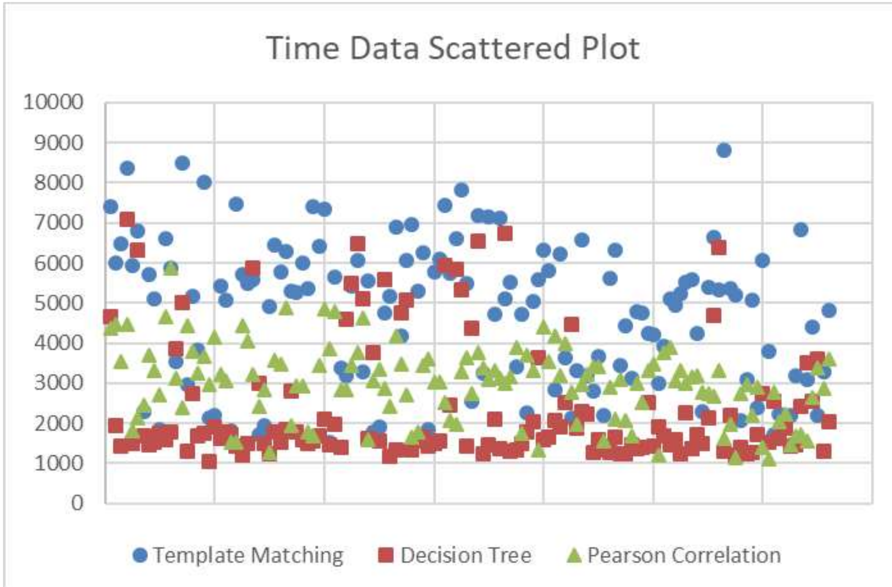


Figure 4.21: Scattered plot of time data by all gestures combined.

accuracy and time performance of Pearson correlation fall in the middle range.

Table 4.7: Result summary of all gestures combined.

| | Template Matching | Decision Tree | Pearson Correlation |
|---|---|---|---|
| Accuracy (%) | 96.2 | 90.2 | 93.9 |
| Average Time (ticks) | 4,770.3 | 2,372.2 | 2,973.3 |

# CHAPTER 5

# DISCUSSION

## 5.1 Template Matching Algorithm

Table 5.1 shows the confusion matrix of the template matching algorithm. As shown in the matrix table, the template matching algorithm performs well on accuracy for the most part.

Table 5.1: Confusion matrix of the template matching algorithm.

| | | Actual Class | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | | RD | RU | LD | LU | DU | DL | Total Predicted |
| Predicted class | RD | 20 | | | | | | 20 |
| | RU | | 20 | | 1 | | | 21 |
| | LD | | | 22 | | | | 22 |
| | LU | | | | 21 | | | 21 |
| | DU | | 2 | | | 22 | | 24 |
| | DL | 2 | | | | | 22 | 24 |
| | Total Actual | 22 | 22 | 22 | 22 | 22 | 22 | |

The precision of a class in a confusion matrix is calculated using the following equation:

$$Precision = \frac{TruePositives}{TruePositives + FalsePositives} = \frac{DiagonalValue}{TotalPredicted} \tag{5.1}$$

Table 5.2: Calculation summary of the template matching algorithm.

| Gesture | Precision | Recall | F-Measure |
|---|---|---|---|
| RD | 1.00 | 0.91 | 0.95 |
| RU | 0.95 | 0.91 | 0.93 |
| LD | 1.00 | 1.00 | 1.00 |
| LU | 1.00 | 0.95 | 0.98 |
| DU | 0.92 | 1.00 | 0.96 |
| DL | 0.92 | 1.00 | 0.96 |

The recall of a class in a confusion matrix is calculated using the following equation:

$$Recall = \frac{TruePositives}{TruePositives + FalseNegatives} = \frac{DiagonalValue}{TotalActual} \tag{5.2}$$

The F-measure, also known as the harmonic mean of precision and recall, is calculated using the following equation:

$$F = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{5.3}$$

Table 5.2 shows a result summary of the calculated precision, recall, and F-measure of each gesture type based on the data provided in Table 5.1 for the template matching algorithm.

The accuracy of the template matching algorithm is calculated as:

$$\begin{aligned} ACC_{TM} &= \frac{TruePositives + TrueNegatives}{ConditionPositives + ConditionNegatives} \\ &= \frac{20 + 20 + 22 + 21 + 22 + 22}{22 \times 6} = \frac{127}{132} = 96.2\% \end{aligned} \tag{5.4}$$

## 5.2 Decision Tree Algorithm

Table 5.3 shows the confusion matrix of the decision tree algorithm. As shown in the matrix table, the decision tree algorithm performs fairly well but makes mistakes sometimes. It appears that mistakes most often take place for gesture types with the same general direction, which makes sense considering the decision tree algorithm is dependent on the calculated features such as the start and end point, slope and length of the bounding box diagonal, and the area of the bounding box.

Table 5.3: Confusion matrix of the decision tree algorithm.

| | | Actual Class | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | | RD | RU | LD | LU | DU | DL | Total Predicted |
| Predicted class | RD | 18 | | | | | 3 | 21 |
| | RU | 2 | 19 | | | 3 | | 24 |
| | LD | | | 22 | | | | 22 |
| | LU | | | | 22 | | | 22 |
| | DU | 1 | 3 | | | 19 | | 23 |
| | DL | 1 | | | | | 19 | 20 |
| | Total Actual | 22 | 22 | 22 | 22 | 22 | 22 | |

Table 5.4 shows a result summary of the calculated precision, recall, and F-measure of each gesture type based on the data provided in Table 5.3 for the decision tree algorithm.

The accuracy of the decision tree algorithm is calculated as:

$$
\begin{aligned}
ACC_{DT} &= \frac{TruePositives + TrueNegatives}{ConditionPositives + ConditionNegatives} \\
&= \frac{18+19+22+22+19+19}{22 \times 6} = \frac{119}{132} = 90.2\%
\end{aligned}
\tag{5.5}
$$

Table 5.4: Calculation summary of the decision tree algorithm.

| Gesture | Precision | Recall | F-Measure |
|---|---|---|---|
| RD | 0.86 | 0.82 | 0.84 |
| RU | 0.79 | 0.86 | 0.83 |
| LD | 1.00 | 1.00 | 1.00 |
| LU | 1.00 | 1.00 | 1.00 |
| DU | 0.83 | 0.86 | 0.84 |
| DL | 0.95 | 0.86 | 0.90 |

## 5.3 Pearson Correlation Algorithm

Table 5.5 shows the confusion matrix of the Pearson Correlation algorithm. As shown in the matrix table, Pearson correlation performs well for most gestures except for the right-up gesture type, where it frequently false classifies it as the diagonal-upper gesture type. It is also worth noting that both the false positives of the right-down gesture type are the diagonal-down gesture type.

Table 5.5: Confusion matrix of the Pearson correlation algorithm

| | | Actual Class | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | | RD | RU | LD | LU | DU | DL | Total Predicted |
| Predicted class | RD | 20 | | | | | | 20 |
| | RU | | 16 | | | | | 16 |
| | LD | | | 22 | | | | 22 |
| | LU | | | | 22 | | | 22 |
| | DU | | 5 | | | 22 | | 27 |
| | DL | 2 | 1 | | | | 22 | 25 |
| | Total Actual | 22 | 22 | 22 | 22 | 22 | 22 | |

Table 5.6 shows a result summary of the calculated precision, recall, and F-measure of each gesture type based on the data provided in Table 5.5 for the Pearson correlation algorithm.

Table 5.6: Calculation summary of the Pearson correlation algorithm.

| Gesture | Precision | Recall | F-Measure |
|---|---|---|---|
| RD | 1.00 | 0.91 | 0.95 |
| RU | 1.00 | 0.73 | 0.84 |
| LD | 1.00 | 1.00 | 1.00 |
| LU | 1.00 | 1.00 | 1.00 |
| DU | 0.81 | 1.00 | 0.90 |
| DL | 0.88 | 1.00 | 0.94 |

The accuracy of the Pearson correlation algorithm is calculated as:

$$
\begin{aligned}
ACC_{PC} &= \frac{TruePositives + TrueNegatives}{ConditionPositives + ConditionNegatives} \\
&= \frac{20 + 16 + 22 + 22 + 22 + 22}{22 \times 6} = \frac{124}{132} = 93.9\%
\end{aligned}
\tag{5.6}
$$

## 5.4 Effect of Input Gestures

Depending on the type of input gesture, certain algorithm may perform better on performance. The input gesture type seems to have an effect on the accuracy performance of the decision tree algorithm. When two gesture types have similar general moving direction, the decision tree algorithm is found to make the most mistakes. Both the template matching algorithm and the Pearson correlation algorithm perform well on classifying gesture types, except for occasional mistakes on the right-down and the right-up gestures. It is worth noting that the false positives of the right-down gesture and the right-up gesture are frequently found to be the diagonal-lower gesture and the diagonal-upper gesture. A possible source of error might be that not enough data points are collected for these gestures, which would result in fewer input data after the re-sampling stage and ultimately lead to false classification. As for time measurement, it is largely dependent on the hardware and the operating system, which could result in inconsistent time readings.

## 5.5 Effect of Base Gestures

The results are directly dependent on the number of base gestures in the system. With more number of gestures being added, the template matching algorithm would still continue to be accurate in recognizing gestures, but the recognition time will increase correspondingly. For the decision tree algorithm and the Pearson correlation algorithm, there may not be a significant increase in time performance as the number of base gestures increases, but the recognition accuracy would reduce.

# CHAPTER 6

# FUTURE WORK

## 6.1 Challenges to be Addressed

In this research, most of the performance results are expected. The classification algorithms faced some difficulties for gesture types with similar directions, which may have been due to not having enough data points. For future work, a larger screen could be used in order to collect more data points. The implementation methods of the algorithms may also have an effect on the classification accuracy. One can study how different ways of algorithm implementation can affect performance and examine why different gestures affect performance.

Another challenge faced during this study was inconsistent user behavior. Some users responded well to eye-tracking and others experienced difficulty. There were also instances where a user was constantly readjusting and the calibration process was redone multiple times throughout testing. In order to mitigate these human errors, one can increase the user study size and collect more samples. Another way to reduce inconsistent user behavior could be increasing the user study length until satisfying user input is obtained. However, this could potentially cause the user to experience more fatigue, which would naturally reduce the quality of user input.

## 6.2 Potential Next Steps

In this research, three classification machine language algorithms were implemented and studied. Similar studies can be done on different algorithms. For example, the neural network algorithm, a machine learning algorithm that simulates the functioning of human brain and computes the output from inputs with associated weights [58], is also a popular choice for eye-tracking [59, 60] and other applications. To continue this research, the next step would be the implementation of using the neural network algorithm to detect gaze input and analyze its performance. There are different kinds of neural network algorithms [61] as well, such as the Levenberg-Marquardt learning algorithm [62, 63], the back propagation learning algorithm [64], and the perceptron learning algorithm [65], and so on. Different kinds of neural networks can be studied and compared for gaze recognition performance.

# CHAPTER 7

# CONCLUSION

Overall, on accuracy performance, the template matching algorithm has the best performance, followed by the Pearson correlation algorithm, and lastly the decision tree algorithm. On timing performance, the decision tree algorithm takes the least amount of the time, closely followed by the Pearson correlation algorithm, and lastly the template matching algorithm. The results are directly dependent on the number of base gestures in the system. As the number of base gestures increases, the template matching algorithm would still maintain its accuracy in recognizing gestures, but the recognition time will increase. For the decision tree algorithm and the Pearson correlation algorithm, as the number of base gestures increases, the recognition accuracy would reduce without a significant increase in time performance. When it comes to deciding which algorithm to use, one might want to consider the properties of the input gesture type and the nature of the application. The template matching algorithm is recommended for accuracy-demanding situations. The decision tree algorithm and the Pearson correlation algorithm is recommended for time-demanding situations but one should be cautious about the possibility that similar gesture types may have similar properties. The algorithms are recommended for when the input gesture types are vastly different, which would help the algorithms achieve higher accuracy.

# REFERENCES


[1] R. J. Jacob and K. S. Karn, "Commentary on section 4 - eye tracking in human-computer interaction and usability research: Ready to deliver the promises," in *The Mind's Eye* (J. Hyönä, R. Radach, and H. Deubel, eds.), pp. 573–605, Amsterdam, NL: North-Holland, 2003.

[2] L. E. Sibert and R. J. K. Jacob, "Evaluation of eye gaze interaction," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '00, (New York, NY, USA), pp. 281–288, ACM, 2000.

[3] R. J. K. Jacob, "What you look at is what you get: Eye movement-based interaction techniques," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '90, (New York, NY, USA), pp. 11–18, ACM, 1990.

[4] D. Fono and R. Vertegaal, "Eyewindows: Evaluation of eye-controlled zooming windows for focus selection," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '05, (New York, NY, USA), pp. 151–160, ACM, 2005.

[5] V. Rajanna and T. Hammond, "Gawschi: Gaze-augmented, wearable-supplemented computer-human interaction," in *Proceedings of the Ninth Biennial ACM Symposium on Eye Tracking Research & Applications*, ETRA '16, (New York, NY, USA), pp. 233–236, ACM, 2016.

[6] S. Zhai, C. Morimoto, and S. Ihde, "Manual and gaze input cascaded (magic) pointing," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '99, (New York, NY, USA), pp. 246–253, ACM, 1999.

[7] E. Castellina, F. Corno, and P. Pellegrino, "Integrated speech and gaze control for realistic desktop environments," in *Proceedings of the 2008 Symposium on Eye Tracking Research & Applications*, ETRA '08, (New York, NY, USA), pp. 79–82, ACM, 2008.

[8] V. D. Rajanna, "Gaze and foot input: Toward a rich and assistive interaction modality," in *Companion Publication of the 21st International Conference on Intelligent User Interfaces*, IUI '16 Companion, (New York, NY, USA), pp. 126–129, ACM, 2016.

[9] V. Rajanna and T. Hammond, "A gaze gesture-based paradigm for situational impairments, accessibility, and rich interactions," in *Proceedings of the Tenth Biennial ACM Symposium on Eye Tracking Research and Applications*, ETRA '18, (New York, NY, USA), ACM, 2018.

[10] D. W. Hansen, H. H. T. Skovsgaard, J. P. Hansen, and E. Møllenbach, "Noise tolerant selection by gaze-controlled pan and zoom in 3d," in *Proceedings of the 2008 Symposium on Eye Tracking Research & Applications*, ETRA '08, (New York, NY, USA), pp. 205–212, ACM, 2008.

[11] P. Majaranta and K.-J. Räihä, "Twenty years of eye typing: Systems and design issues," in *Proceedings of the 2002 Symposium on Eye Tracking Research & Applications*, ETRA '02, (New York, NY, USA), pp. 15–22, ACM, 2002.

[12] J. P. Hansen, A. S. Johansen, D. W. Hansen, K. Itoh, and S. Mashino, "Command without a click: Dwell time typing by mouse and gaze selections," in *Proceedings of the International Conference on Human-Computer Interaction*, INTERACT '03, (Zürich, Switzerland), pp. 121–128, IOS Press, 2003.

[13] D. W. Hansen, J. P. Hansen, M. Niels, and M. B. Stegmann, "Eye typing using markov and active appearance models," in *Proceedings of the Sixth IEEE Workshop on Applications of Computer Vision*, WACV '02, (Washington, DC, USA), pp. 132–137, IEEE Computer Society, 2002.

[14] V. Rajanna and J. P. Hansen, "Gaze typing in virtual reality: Impact of keyboard design, selection method, and motion," in *Proceedings of the Tenth Biennial ACM Symposium on Eye Tracking Research and Applications*, ETRA '18, (New York, NY, USA), ACM, 2018.

[15] S. Stellmach and R. Dachselt, "Look & touch: Gaze-supported target acquisition," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '12, (New York, NY, USA), pp. 2981–2990, ACM, 2012.

[16] P. Majaranta and A. Bulling, "Eye tracking and eye-based humanâĂŞcomputer interaction," in *Advances in Physiological Computing*, Springer, London, 2014.

[17] J. P. Hansen, V. Rajanna, I. S. MacKenzie, and P. Bæ kgaard, "A fitts' law study of click and dwell interaction by gaze, head and mouse with a head-mounted display," in *COGAIN '18: Workshop on Communication by Gaze Interaction, June 14–17, 2018, Warsaw, Poland*, COGAIN '18, (New York, NY, USA), ACM, 2018.

[18] A. Bulling, F. Alt, and A. Schmidt, "Increasing the security of gaze-based cued-recall graphical passwords using saliency masks," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '12, (New York, NY, USA), pp. 3011–3020, ACM, 2012.

[19] J. Weaver, K. J. Mock, and B. Hoanca, "Gaze-based password authentication through automatic clustering of gaze points," in *2011 IEEE International Conference on Systems, Man, and Cybernetics*, SMC '11, (Washington, DC, USA), pp. 2749–2754, IEEE, 2011.

[20] A. Hyrskykari, H. Istance, and S. Vickers, "Gaze gestures or dwell-based interaction?," in *Proceedings of the Symposium on Eye Tracking Research and Applications*, ETRA '12, (New York, NY, USA), pp. 229–232, ACM, 2012.

[21] J. D. Smith and T. C. N. Graham, "Use of eye movements for video game control," in *Proceedings of the 2006 ACM SIGCHI International Conference on Advances in Computer Entertainment Technology*, ACE '06, (New York, NY, USA), ACM, 2006.

[22] P. Kaul, V. Rajanna, and T. Hammond, "Exploring users' perceived activities in a sketch-based intelligent tutoring system through eye movement data," in *Proceedings of the ACM Symposium on Applied Perception*, SAP '16, (New York, NY, USA), pp. 134–134, ACM, 2016.

[23] B. Bauman, R. Gunhouse, A. Jones, W. Da Silva, S. Sharar, V. Rajanna, J. Cherian, J. I. Koh, and T. Hammond, "Visualeyeze: A web-based solution for receiving feedback on artwork through eye tracking," in *IUI 2018 Workshop on Web Intelligence and Interaction*, 2018.

[24] C. Hennessey and J. Fiset, "Long range eye tracking: Bringing eye tracking into the living room," in *Proceedings of the Symposium on Eye Tracking Research and Applications*, ETRA '12, (New York, NY, USA), pp. 249–252, ACM, 2012.

[25] 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.

[26] T.-S. Lim, W.-Y. Loh, and Y.-S. Shih, "A comparison of prediction accuracy, complexity, and training time of thirty-three old and new classification algorithms," *Machine Learning*, vol. 40, pp. 203–228, sep 2000.

[27] V. Park and M. S. Carson, "A performance comparison of the temporally-ordered routing algorithm and ideal link-state routing," in *Proceedings of the Third IEEE Symposium on Computers and Communications*, ISCC '98, (Washington, DC, USA), pp. 592–598, IEEE, jun 1998.

[28] J. O. Wobbrock, A. D. Wilson, and Y. Li, "Gestures without libraries, toolkits or training: A $1 recognizer for user interface prototypes," in *Proceedings of the 20th Annual ACM Symposium on User Interface Software and Technology*, UIST '07, (New York, NY, USA), pp. 159–168, ACM, 2007.

[29] D. Rubine, "Specifying gestures by example," *SIGGRAPH Comput. Graph.*, vol. 25, pp. 329–337, jul 1991.

[30] M. Vidal, A. Bulling, and H. Gellersen, "Pursuits: Spontaneous interaction with displays based on smooth pursuit eye movement and moving targets," in *Proceedings of the 2013 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, UbiComp '13, (New York, NY, USA), pp. 439–448, ACM, 2013.

[31] M.-C. Su, K.-C. Wang, and G.-D. Chen, "An eye tracking system and its application in aids for people with severe disabilities," *Biomedical Engineering: Applications, Basis and Communications*, vol. 18, no. 6, pp. 319–327, 2006.

[32] V. Rajanna, "Gaze typing through foot-operated wearable device," in *Proceedings of the 18th International ACM SIGACCESS Conference on Computers and Accessibility*, ASSETS '16, (New York, NY, USA), pp. 345–346, ACM, 2016.

[33] V. Rajanna, A. H. Malla, R. A. Bhagat, and T. Hammond, "Dygazepass: A gaze gesture-based dynamic authentication system to counter shoulder surfing and video analysis attacks," in *2018 IEEE 4th International Conference on Identity, Security, and Behavior Analysis (ISBA)*, ISBA '18, (Washington, DC, USA), pp. 1–8, IEEE, 2018.

[34] 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.

[35] S. Djamasbi, M. Siegel, and T. Tullis, "Generation y, web design, and eye tracking," *Int. J. Hum.-Comput. Stud.*, vol. 68, pp. 307–323, may 2010.

[36] K. Mock, B. Hoanca, J. Weaver, and M. Milton, "Real-time continuous iris recognition for authentication using an eye tracker," in *Proceedings of the 2012 ACM Conference on Computer and Communications Security*, CCS '12, (New York, NY, USA), pp. 1007–1009, ACM, 2012.

[37] A. Duchowski, *Eye Tracking Methodology: Theory and Practice*. London, England, UK: Springer-Verlag London, second ed., 2007.

[38] A. T. Duchowski, "A breadth-first survey of eye-tracking applications," *Behavior Research Methods, Instruments, & Computers*, vol. 32, pp. 455–470, nov 2002.

[39] L. Stark, G. Vossius, and L. R. Young, "Predictive control of eye tracking movements," *IRE Transactions on Human Factors in Electronics*, vol. HFE-3, pp. 52–57, oct 1962.

[40] S. Jayaram, J. Vance, R. Gadh, U. Jayaram, and H. Srinivasan, "Assessment of vr technology and its applications to engineering problems," *Journal of Computing and Information Science in Engineering*, vol. 1, pp. 72–83, jan 2001.

[41] V. Rajanna and T. Hammond, "A fitts' law evaluation of gaze input on large displays compared to touch and mouse inputs," in *COGAIN '18: Workshop on Communication by Gaze Interaction, June 14–17, 2018, Warsaw, Poland*, COGAIN '18, (New York, NY, USA), ACM, 2018.

[42] Y. Zhang, A. Bulling, and H. Gellersen, "Sideways: A gaze interface for spontaneous interaction with situated displays," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '13, (New York, NY, USA), pp. 851–860, ACM, 2013.

[43] C. H. Morimoto and M. R. Mimica, "Eye gaze tracking techniques for interactive applications," *Computer Vision and Image Understanding*, vol. 98, no. 1, pp. 4–24, 2005.

[44] M. E. Holmes, S. Josephson, and R. E. Carney, "Visual attention to television programs with a second-screen application," in *Proceedings of the Symposium on Eye Tracking Research and Applications*, ETRA '12, (New York, NY, USA), pp. 397–400, ACM, 2012.

[45] K. Talmi and J. Liu, "Eye and gaze tracking for visually controlled interactive stereoscopic displays," *Signal Processing: Image Communication*, vol. 14, pp. 799–810, aug 1999.

[46] Z. Zhu and Q. Ji, "Eye and gaze tracking for interactive graphic display," *Machine Vision and Applications*, vol. 15, pp. 139–148, jul 2004.

[47] J. Nielsen and K. Pernice, *Eyetracking Web Usability*. Thousand Oaks, CA, USA: New Riders Publishing, first ed., 2009.

[48] B. Pan, H. A. Hembrooke, G. K. Gay, L. A. Granka, M. K. Feusner, and J. K. Newman, "The determinants of web page viewing behavior: An eye-tracking study," in *Proceedings of the 2004 Symposium on Eye Tracking Research & Applications*, ETRA '04, (New York, NY, USA), pp. 147–154, ACM, 2004.

[49] M. Wedel and R. Pieters, "A review of eye-tracking research in marketing," in *Review of Marketing Research*, vol. 4, ch. 5, pp. 123–147, Emerald Group Publishing Limited, 2008.

[50] J. D. VeláSquez, "Combining eye-tracking technologies with web usage mining for identifying website keyobjects," *Eng. Appl. Artif. Intell.*, vol. 26, pp. 1469–1478, may 2013.

[51] R. Johnson, K. O'Hara, A. Sellen, C. Cousins, and A. Criminisi, "Exploring the potential for touchless interaction in image-guided interventional radiology," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '11, (New York, NY, USA), pp. 3323–3332, ACM, 2011.

[52] M. Mrochen, M. S. Eldine, M. Kaemmerer, T. Seiler, and W. Hütz, "Improvement in photorefractive corneal laser surgery results using an active eye-tracking system," *J Cataract Refract Surg*, vol. 27, pp. 1000–1006, jul 2001.

[53] N. Ahmidi, M. Ishii, G. Fichtinger, G. L. Gallia, and G. D. Hager, "An objective and automated method for assessing surgical skill in endoscopic sinus surgery using eye-tracking and tool-motion data," *Int Forum Allergy Rhinol*, vol. 2, pp. 507–515, nov 2012.

[54] A. S. A. Chetwood, K.-W. Kwok, L.-W. Sun, G. P. Mylonas, J. Clark, A. Darzi, and G.-Z. Yang, "Collaborative eye tracking: a potential training tool in laparoscopic surgery," *Surgical Endoscopy*, vol. 26, pp. 2003–2009, jul 2012.

[55] V. Rajanna and T. Hammond, "Gaze-assisted user authentication to counter shoulder-surfing attacks," in *Proceedings of the 2016 ACM Richard A. Tapia Celebration of Diversity in Computing*, TAPIA '16, (Austin, TX, USA), p. arXiv:1803.07782, ACM, 2018.

[56] Microsoft, "Stopwatch.elapsedticks property." https://msdn.microsoft.com/en-us/library/system.diagnostics.stopwatch.elapsedticks(v=vs.110).aspx, 2018.

[57] Microsoft, "Stopwatch.frequency field." https://msdn.microsoft.com/en-us/library/system.diagnostics.stopwatch.frequency(v=vs.110).aspx, 2018.

[58] M. T. Hagan, H. B. Demuth, M. H. Beale, and O. D. Jesús, *Neural Network Design*. Stillwater, OK, USA: Martin Hagan, second ed., sep 2014.

[59] W. Sewell and O. Komogortsev, "Real-time eye gaze tracking with an unmodified commodity webcam employing a neural network," in *CHI '10 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '10, (New York, NY, USA), pp. 3739–3744, ACM, 2010.

[60] L. Behera, I. Kar, and A. C. Elitzur, "A recurrent quantum neural network model to describe eye tracking of moving targets," *Foundations of Physics Letters*, vol. 18, pp. 357–370, aug 2005.

[61] Microsoft, "Accord.neuro.learning namespace." https://bit.ly/2Hqb6K7, 2018.

[62] G. Lera and M. Pinzolas, "Neighborhood based levenberg-marquardt algorithm for neural network training," *Trans. Neur. Netw.*, vol. 13, no. 5, pp. 1200–1203, 2002.

[63] B. G. Kermani, S. S. Schiffman, and H. T. Nagle, "Performance of the Levenberg-Marquardt neural network training method in electronic nose applications," *Sensors and Actuators B: Chemical*, vol. 110, pp. 13–22, sep 2005.

[64] R. Hecht-Nielsen, "Theory of the backpropagation neural network," in *Neural Networks for Perception* (H. Wechsler, ed.), vol. 2, pp. 65–93, Orlando, FL, USA: Harcourt Brace & Co., 1992.

[65] S. O. Haykin, *Neural Networks and Learning Machines*. Upper Saddle River, NJ, USA: Pearson Education, Inc., third ed., nov 2008.

# APPENDIX: INPUT GESTURE DATA RESULTS

Table 7.1: Input Gesture Type: Right-Down*

| User | Template Matching | Time (ticks) | Decision Tree | Time (ticks) | Pearson Correlation | Time (ticks) |
|---|---|---|---|---|---|---|
| 1 | RD | 7395 | RD | 4669 | RD | 4371 |
| 2 | RD | 6003 | RD | 1941 | RD | 4458 |
| 3 | RD | 6479 | RD | 1433 | RD | 3534 |
| 4 | RD | 8358 | RD | 7089 | RD | 4478 |
| 5 | RD | 5936 | **DU** | 1494 | RD | 1819 |
| 6 | RD | 6790 | RD | 6319 | RD | 2143 |
| 7 | RD | 2292 | RD | 1674 | RD | 2462 |
| 8 | RD | 5710 | RD | 1458 | RD | 3690 |
| 9 | RD | 5101 | RD | 1512 | RD | 3302 |
| 10 | RD | 1858 | **RU** | 1752 | RD | 2719 |
| 11 | **DL** | 6620 | RD | 1587 | RD | 4664 |
| 12 | RD | 5868 | RD | 1789 | RD | 5885 |
| 13 | RD | 3531 | RD | 3853 | RD | 3124 |
| 14 | RD | 8486 | RD | 5005 | RD | 2386 |
| 15 | RD | 2972 | RD | 1317 | RD | 4445 |
| 16 | RD | 5157 | RD | 2726 | RD | 3810 |
| 17 | RD | 3824 | **RU** | 1696 | RD | 3263 |
| 18 | **DL** | 8020 | RD | 1755 | **DL** | 3671 |
| 19 | RD | 2144 | RD | 1053 | RD | 2959 |
| 20 | RD | 2190 | RD | 1902 | RD | 4140 |
| 21 | RD | 5410 | RD | 1634 | RD | 3205 |
| 22 | RD | 5072 | **DL** | 1798 | **DL** | 3065 |

*Gray-bolded cells were classified incorrectly

Table 7.2: Input Gesture Type: Right-Up*

| User | Template Matching | Time (ticks) | Decision Tree | Time (ticks) | Pearson Correlation | Time (ticks) |
|---|---|---|---|---|---|---|
| 1 | RU | 1811 | RU | 1716 | RU | 1523 |
| 2 | RU | 7481 | RU | 1439 | RU | 1517 |
| 3 | RU | 5703 | RU | 1200 | RU | 4430 |
| 4 | RU | 5497 | RU | 1490 | RU | 4043 |
| 5 | RU | 5581 | **DU** | 5864 | RU | 3216 |
| 6 | RU | 1719 | RU | 3004 | **DU** | 2426 |
| 7 | RU | 1942 | **DU** | 1491 | RU | 2846 |
| 8 | RU | 4907 | RU | 1230 | RU | 1274 |
| 9 | RU | 6445 | RU | 1773 | RU | 3558 |
| 10 | RU | 5782 | RU | 1535 | RU | 3491 |
| 11 | **DU** | 6296 | **DU** | 1843 | **DU** | 4875 |
| 12 | RU | 5303 | RU | 2820 | RU | 1931 |
| 13 | RU | 5276 | RU | 1797 | RU | 2921 |
| 14 | **DU** | 5995 | RU | 1600 | **DU** | 2944 |
| 15 | RU | 5350 | RU | 1499 | RU | 1797 |
| 16 | RU | 7397 | RU | 1549 | RU | 1698 |
| 17 | RU | 6410 | RU | 1700 | **DU** | 3430 |
| 18 | RU | 7332 | RU | 2094 | **DL** | 4840 |
| 19 | RU | 1536 | RU | 1450 | RU | 3859 |
| 20 | RU | 5632 | RU | 1981 | RU | 4794 |
| 21 | RU | 3383 | RU | 1388 | RU | 2823 |
| 22 | RU | 3176 | RU | 4595 | **DU** | 2835 |

*Gray-bolded cells were classified incorrectly

Table 7.3: Input Gesture Type: Left-Down*

| User | Template Matching | Time (ticks) | Decision Tree | Time (ticks) | Pearson Correlation | Time (ticks) |
|---|---|---|---|---|---|---|
| 1 | LD | 5413 | LD | 5490 | LD | 3452 |
| 2 | LD | 6067 | LD | 6490 | LD | 3747 |
| 3 | LD | 3278 | LD | 5105 | LD | 4611 |
| 4 | LD | 5545 | LD | 1613 | LD | 1591 |
| 5 | LD | 1786 | LD | 3777 | LD | 3046 |
| 6 | LD | 1916 | LD | 1543 | LD | 3337 |
| 7 | LD | 4765 | LD | 5593 | LD | 2855 |
| 8 | LD | 5162 | LD | 1180 | LD | 2429 |
| 9 | LD | 6898 | LD | 1337 | LD | 4177 |
| 10 | LD | 4189 | LD | 4768 | LD | 3468 |
| 11 | LD | 6066 | LD | 5065 | LD | 2708 |
| 12 | LD | 6941 | LD | 1338 | LD | 1664 |
| 13 | LD | 5296 | LD | 1545 | LD | 1769 |
| 14 | LD | 6238 | LD | 1642 | LD | 3449 |
| 15 | LD | 1854 | LD | 1423 | LD | 3588 |
| 16 | LD | 5787 | LD | 1494 | LD | 3041 |
| 17 | LD | 6079 | LD | 1570 | LD | 3020 |
| 18 | LD | 7422 | LD | 5936 | LD | 2526 |
| 19 | LD | 5740 | LD | 2446 | LD | 2055 |
| 20 | LD | 6601 | LD | 5842 | LD | 1988 |
| 21 | LD | 7831 | LD | 5327 | LD | 3273 |
| 22 | LD | 5490 | LD | 1419 | LD | 3641 |

*Gray-bolded cells were classified incorrectly

Table 7.4: Input Gesture Type: Left-Up*

| User | Template Matching | Time (ticks) | Decision Tree | Time (ticks) | Pearson Correlation | Time (ticks) |
|---|---|---|---|---|---|---|
| 1 | LU | 2550 | LU | 4363 | LU | 2745 |
| 2 | LU | 7165 | LU | 6556 | LU | 3749 |
| 3 | LU | 3266 | LU | 1240 | LU | 3377 |
| 4 | LU | 7149 | LU | 1462 | LU | 3083 |
| 5 | LU | 4735 | LU | 2096 | LU | 3328 |
| 6 | LU | 7126 | LU | 1379 | LU | 3136 |
| 7 | LU | 5116 | LU | 6730 | LU | 3001 |
| 8 | LU | 5504 | LU | 1297 | LU | 3149 |
| 9 | **RU** | 3399 | LU | 1331 | LU | 3881 |
| 10 | LU | 4733 | LU | 1487 | LU | 1745 |
| 11 | LU | 2276 | LU | 1782 | LU | 3711 |
| 12 | LU | 5032 | LU | 2034 | LU | 3329 |
| 13 | LU | 5572 | LU | 3648 | LU | 1324 |
| 14 | LU | 6303 | LU | 1592 | LU | 4387 |
| 15 | LU | 5793 | LU | 1646 | LU | 3551 |
| 16 | LU | 2844 | LU | 2073 | LU | 4189 |
| 17 | LU | 6219 | LU | 1922 | LU | 3175 |
| 18 | LU | 3635 | LU | 2519 | LU | 3978 |
| 19 | LU | 2119 | LU | 4457 | LU | 2779 |
| 20 | LU | 3315 | LU | 1873 | LU | 1959 |
| 21 | LU | 6570 | LU | 2294 | LU | 2970 |
| 22 | LU | 3189 | LU | 2244 | LU | 3234 |

*Gray-bolded cells were classified incorrectly

Table 7.5: Input Gesture Type: Diagonal-Upper*

| User | Template Matching | Time (ticks) | Decision Tree | Time (ticks) | Pearson Correlation | Time (ticks) |
|---|---|---|---|---|---|---|
| 1 | DU | 2817 | **RU** | 1278 | DU | 3410 |
| 2 | DU | 3679 | DU | 1602 | DU | 3412 |
| 3 | DU | 2199 | DU | 1322 | DU | 1569 |
| 4 | DU | 5604 | **RU** | 1278 | DU | 2897 |
| 5 | DU | 6305 | DU | 1647 | DU | 2105 |
| 6 | DU | 3430 | DU | 1229 | DU | 3088 |
| 7 | DU | 4442 | DU | 1232 | DU | 2064 |
| 8 | DU | 3113 | DU | 1447 | DU | 1700 |
| 9 | DU | 4781 | DU | 1382 | DU | 2999 |
| 10 | DU | 4746 | DU | 1401 | DU | 2518 |
| 11 | DU | 4227 | DU | 2523 | DU | 3317 |
| 12 | DU | 4201 | DU | 1415 | DU | 3464 |
| 13 | DU | 2986 | DU | 1916 | DU | 1223 |
| 14 | DU | 3919 | DU | 1685 | DU | 3750 |
| 15 | DU | 5109 | DU | 1460 | DU | 3889 |
| 16 | DU | 4957 | DU | 1605 | DU | 3058 |
| 17 | DU | 5221 | **RU** | 1230 | DU | 3307 |
| 18 | DU | 5508 | DU | 2262 | DU | 3003 |
| 19 | DU | 5598 | DU | 1368 | DU | 3145 |
| 20 | DU | 4257 | DU | 1710 | DU | 3181 |
| 21 | DU | 2296 | DU | 1489 | DU | 2774 |
| 22 | DU | 5378 | DU | 2144 | DU | 2743 |

*Gray-bolded cells were classified incorrectly

Table 7.6: Input Gesture Type: Diagonal-Lower*

| User | Template Matching | Time (ticks) | Decision Tree | Time (ticks) | Pearson Correlation | Time (ticks) |
|---|---|---|---|---|---|---|
| 1 | DL | 6626 | DL | 4674 | DL | 2685 |
| 2 | DL | 5320 | DL | 6393 | DL | 3321 |
| 3 | DL | 8817 | DL | 1313 | DL | 1617 |
| 4 | DL | 5352 | **RD** | 2182 | DL | 1976 |
| 5 | DL | 5206 | DL | 1221 | DL | 1128 |
| 6 | DL | 2054 | DL | 1405 | DL | 2729 |
| 7 | DL | 3081 | DL | 1254 | DL | 2968 |
| 8 | DL | 5077 | DL | 1271 | DL | 2192 |
| 9 | DL | 2401 | DL | 1707 | DL | 2910 |
| 10 | DL | 6059 | **RD** | 2732 | DL | 1400 |
| 11 | DL | 3806 | DL | 1527 | DL | 1127 |
| 12 | DL | 1711 | DL | 2550 | DL | 2761 |
| 13 | DL | 2230 | DL | 1620 | DL | 2028 |
| 14 | DL | 1958 | DL | 1890 | DL | 2222 |
| 15 | DL | 2205 | DL | 1415 | DL | 1454 |
| 16 | DL | 3202 | DL | 1459 | DL | 1677 |
| 17 | DL | 6844 | DL | 2413 | DL | 1707 |
| 18 | DL | 3104 | DL | 3505 | DL | 1557 |
| 19 | DL | 4411 | DL | 2544 | DL | 2651 |
| 20 | DL | 2198 | DL | 3618 | DL | 3383 |
| 21 | DL | 3271 | **RD** | 1299 | DL | 2873 |
| 22 | DL | 4832 | DL | 2035 | DL | 3608 |

*Gray-bolded cells were classified incorrectly