

Article

Examining Factors Influencing Cognitive Load of Computer Programmers

Didem Issever ^{1,*}, Mehmet Cem Catalbas ² and Fecir Duran ¹¹ Department of Computer Engineering, Faculty of Technology, Gazi University, 06560 Ankara, Turkey; fduran@gazi.edu.tr² Department of Electronics and Automation, 1st Organized Industrial Zone Vocational School, Ankara University, 06935 Ankara, Turkey; mccatalbas@ankara.edu.tr

* Correspondence: didem.issever@gazi.edu.tr

Abstract: In this study, the factors influencing the cognitive load of computer programmers during the perception of different code tasks were investigated. The eye movement features of computer programmers were used to provide a significant relationship between the perceptual processes of the sample codes and cognitive load. Thanks to the relationship, the influence of various personal characteristics of programmers on cognitive load was examined. Various personal parameters such as programming experience, age, native language, and programming frequency were used in the study. The study was performed on the Eye Movements in Programming (EMIP) dataset containing 216 programmers with different characteristics. Eye movement information recorded during two different code comprehension tasks was decomposed into sub-information, such as pupil movement speed and diameter change. Rapid changes in eye movement signals were adaptively detected using the z-score peak detection algorithm. Regarding the cognitive load calculations, canonical correlation analysis was used to build a statistically significant and efficient mathematical model connecting the extracted eye movement features and the different parameters of the programmers, and the results were statistically significant. As a result of the analysis, the factors affecting the cognitive load of computer programmers for the related database were converted into percentages, and it was seen that linguistic distance is an essential factor in the cognitive load of programmers and the effect of gender on cognitive load is quite limited.

Citation: Issever, D.; Catalbas, M.C.; Duran, F. Examining Factors Influencing Cognitive Load of Computer Programmers. *Brain Sci.* **2023**, *13*, 1132. <https://doi.org/10.3390/brainsci13081132>

Academic Editors: David Facal and Gerry Leisman

Received: 16 June 2023

Revised: 26 July 2023

Accepted: 27 July 2023

Published: 28 July 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

Keywords: saccadic eye movement; cognitive load; eye tracking; computer programming; canonical correlation; linguistic distance

1. Introduction

Saccadic eye movements are quick movements that occur when visual focus on an object changes rapidly [1]. There are strong connections between saccadic eye movements and cognition levels. It has been found that it is meaningful to analyze not only saccadic eye movements but also pupillary parameters to determine cognitive states; with suitable combinations of these variables, the cognitive load levels of individuals for specific activities can be determined [2]. Therefore, many studies have analyzed the cognitive levels of individuals using eye movements for different tasks [3]. The measurement of rapid eye movements, such as saccadic eye movements, has improved with the performance of eye-tracking devices, and the success and number of studies in this field have increased significantly in recent years. In this article, some of these studies are discussed. Pfleging et al. conducted a fully controlled experiment with an intuitive classifier under different illumination conditions, during which a dataset and preliminary model were developed to estimate the mental load level based on eye movement [4]. In a driving simulator study, Kun et al. analyzed the pupil diameter measurements of a distant speaker and a driver preparing to speak and discovered that the pupil diameter of the driver was more

significant than that of the remote speaker as the driver was preparing to speak. They also investigated the effects of different conversations on the pupil diameter. Consequently, they aimed to create a dialogue system capable of adapting to the driver's behavior under a high cognitive load [5].

Kruger and colleagues used the percentage change in pupil diameters to examine the cognitive load that occurs when participants watch a video with and without subtitles. They discovered cognitive load decreased in courses with subtitles when watching videos [6]. Chen et al. suggested an intelligent cognitive load monitoring system based on the use of the eyes. According to the proposed method, the cognitive load was created using arithmetic exercises and controlled by the number of moves and digits. They used regression models to identify more than two cognitive load levels using different eye activity models. It was observed that the pupil diameter and number of blinks were found to increase more during complicated tasks [7]. Zagermann et al. proposed employing eye-tracking tools to study users' cognitive load when interacting with a system. In this context, they provided a model that integrates human–computer interaction features into the link between eye-tracking data and cognitive load. They believe that this approach will help develop interfaces that require less cognitive capacity and suggest that eye tracking is a significant tool for examining cognitive processes in visual computation [8]. In another study, Cong et al. examined the relationship between cognitive load and eye movements. They investigated cognitive load in the multimedia presentation process. They developed a quantitative model to assess cognitive load in terms of knowledge comprehension and used pupil diameter variation for an assessment focused on the cognitive load theory [9].

The term cognition refers to the ability of the brain to collect, process, and convert information received from various sources such as sensory stimuli, perception, experience, and vision. Cognitive load theory aims to objectively analyze and make sense of the human brain's perception, understanding, and learning processes [10]. This exciting theory is used in many disciplines, especially education. Although it is believed that there are evolutionary similarities in information processing in individuals, it should be remembered that the ability to store information varies from person to person and is influenced by a variety of factors [11]. John Sweller developed cognitive load theory (CLT), and an article on the subject was published in the journal *Cognitive Science* in 1988 [12]. John Sweller revised this theory in an article published in 2019 owing to developments in the theory over time. CLT aims to aid scientists in developing unique instructional solutions compatible with the limitations of the human cognitive system. Working memory, or short-term memory, has a finite capacity and can only effectively process a limited amount of information instantly. This assumption regarding memory capacity for processing information is the basis of the cognitive load theory. Cognitive load theory aims to build meaningful learning design concepts that focus on the human cognitive system. According to cognitive load theory, human cognitive architecture assumes cognitive schemes that contain limited working memory and unlimited long-term memory; expertise is obtained only from knowledge stored in long-term memory as structures. Cognitive load is a multidimensional concept that affects a learner's cognitive system while performing a task. The framework of Paas and van Merrinboer's broad CLT model contains a causal dimension that reflects the interaction of task and learner characteristics and an evaluative dimension that reflects the concepts of measurable load, effort, and mental performance [13,14]. Every stimulus the individual is exposed to while in the learning position creates a load on the individual's mind, and these stimuli may be in the environment in a desired or undesirable manner.

According to CLT, there are three types of cognitive load: intrinsic, extraneous, and germane [15]. Intrinsic load is a cognitive load produced by the complexity or difficulty of learning new knowledge. Researchers in this field have found that learning a structure with many interaction components is more complex than learning a system with fewer interaction elements and requires more cognitive capacity to process [16]. The learning technique cannot change the intrinsic load; it can only change with the degree of skill.

Expert learners with extensive knowledge can quickly combine complex information items with pre-existing schemas and manipulate schema development as a working memory item. Therefore, individuals who are experts in their field have a low intrinsic load, despite solving complex problems. According to CLT, there are two types of loads apart from the intrinsic load [17]. Extraneous load refers to the mental resources allocated to objects that do not contribute to learning and schema creation. This part of cognitive load is about the presentation of information and the instructional format, which can increase the total cognitive load while having little effect on learning; it is the amount of memory consumed by all the hidden programs operating in the background in the system tray [18]. Germane load represents partitioned mental resources for creating and organizing long-term memory schemas, and it is similar to memory usage when loading a program on a computer. In this context, germane load defines the activity of long-term storage of information or schemas, and the peculiarity of this section significantly accelerates the learning process. Germane load enables the development of cognitive behavioral patterns necessary to make sense of the categories of information. As a result of the aggregation of these three loads, cognitive load emerges. This also means that the total cognitive load an individual experiences in working memory while performing a specific task is the sum of the three different load types.

Learning is connected to increasing the processing capacity of the working memory. The learning process involves transferring information from working memory to long-term memory. As in Schema theory, this information transmission allows information to be structured as a schema in long-term memory. Creating a schema entails connecting disparate pieces of information to move from a lower to a higher level of complexity and keeping them together as a single meaningful whole chunk of information. Figure 1 illustrates a mental architecture model and the function of CLT in working memory and schema formation [18].

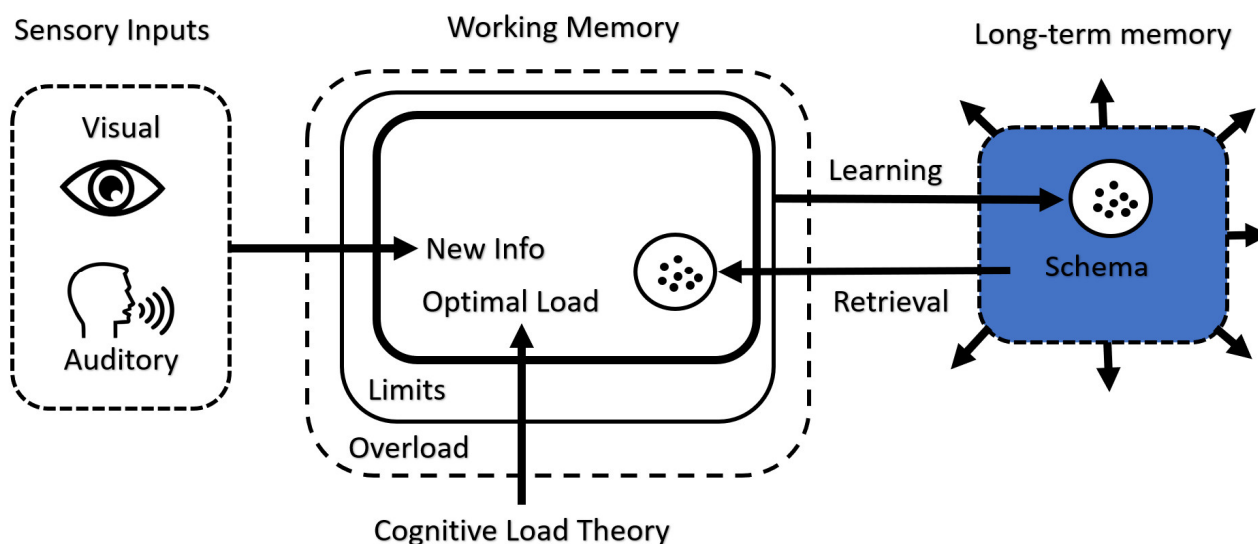


Figure 1. CLT and schema representation.

Measuring this load is as important as defining the concept of a cognitive load. Many methodologies and measurement tools can be used to determine the cognitive load. The following are some approaches for evaluating cognitive load frequently utilized in the literature.

- Emotional Scales: Participants are administered questionnaires assessing the difficulty and confusion level of cognitive tasks.
- Performance Measures: Performance measures such as the time to complete a task, level of accuracy, or number of errors can be indicators of cognitive load.

- **Psychophysiological Measures:** Physiological measurements such as electroencephalogram (EEG), electrocardiogram (ECG), eye movements, or skin conductance can be used to determine cognitive load.

Among these approaches, studies using physiological measurements have increased in popularity in recent years due to their avoidance of subjectivity and the development of sensor technologies. Eye-tracking systems stand out among various physiological parameters for estimating cognitive load. There are various reasons for this. The stimulus is usually presented to the user visually, with the result that there is a significant change in eye movements; there is no physical connection between eye tracking systems and the user; eye movements can be interpreted more easily than other physiological signal measurements; the cost of an eye-tracking system is as much as a comprehensive EEG device or magnetoencephalography (MEG) device; and its application does not require any pre-processing that requires expertise. There are many studies on the calculation of cognitive load from eye movements. In a study conducted in this context, pupil responses were used to establish a meaningful relationship between cognitive load and the difficulty level of a video game [19]. In another study that examined the behavior of map users in the process of performing specific tasks using the relationship between cognitive load and eye movements, significant differences were observed between the eye movements of experts and novice users. As a result of the study, novices were observed to have longer fixation durations and higher saccade velocity, which indicates a higher cognitive load for novices [20]. Another comprehensive study in this field also found a positive correlation between pupil dilation and cognitive load [21]. In another study, the effects of visual and auditory stimuli on cognitive load were analyzed using eye movements. In a related study, the impact of background music (BGM) on cognitive load in learning processes was analyzed through eye movements. It was found that listening to BGM imposed a higher cognitive load on post-lexical processes than on lexical processes [22]. Eye movement and pupillary response indicators of cognitive load, commonly used in studies, are listed below.

- Pupillary diameter mean
- Pupillary diameter deviation
- Saccadic eye movement
- Pupillary hippus

Saccadic eye movement is a type of rapid eye movement in which the eye shifts rapidly from one focal point to another. It is an essential component of visual perception and is controlled by a complex network of neuronal circuits in the brain [23]. During a saccade, the eyes move rapidly from one point to another, followed by a brief fixation that allows the brain to interpret visual information. Because saccadic eye movement is a form of eye movement at high speeds, observing with cameras with high sampling rates is more meaningful and efficient. Saccadic eye movements are essential in various tasks and have been extensively studied in cognitive psychology, neurology, and ophthalmology.

The main objective of this research is to identify the variables that influence programmers' cognitive load during computer programming activities. An extensive database collecting quantitative and categorical information on programmers' spoken language, programming experience, and age was used. Through linguistic distance, categorical information in the database, such as native language, was transformed into quantitative variables, and the database of 216 participants was made appropriate for study through numerical values. As a result of the study, the factors influencing the cognitive load of programmers were determined and detailed in percentages.

2. Materials and Methods

A multivariate analysis method was required to develop the necessary mathematical model within this study's scope. Multivariate analysis methods create a logical relationship between datasets that contain many observations or variables [24]. Creating a meaningful pattern among this large number of variables provides an efficient mathematical

model for solving problems and understanding the main pattern of the dataset. Feature extraction, economics, medicine, and pharmacy sciences are a few application areas of multivariate analytic analysis techniques [25,26]. Because the dataset produced in this study contains numerous variables, an effective multivariate analysis method should be implemented. Therefore, the canonical correlation analysis approach was selected, which is frequently used in multivariate analysis methods and is helpful in understanding and evaluating their outputs.

2.1. Canonical Correlation Analysis

Canonical correlation analysis (CCA) is an effective multivariate analysis tool for determining and evaluating the relationship between two multivariate variable sets [27]. Additionally, it can be used as a dimension-reduction application, such as a principal component or linear discriminant analysis. The main aim of CCA is to discover canonical coefficients that maximize the linear relationships between two datasets [28]. H. Hotelling proposed and published CCA in the literature in 1936, and it has since been used in statistics, biometrics, meteorology, epidemiology, and many other disciplines [29]. CCA is often used in various sectors, including medicine, education, economics, and engineering, to explore and comprehend the relationships between observations. It will be utilized in this study to analyze the importance of the relationship between two different datasets as well as the ratio of the sub-components that influence this relationship. One of these datasets contains cognitive load factors, while the other includes programmers' personal parameters. The basic form of multivariate regression analysis is performed using the coefficient values that provide the best linear relationship between the input parameters x_1, x_2, \dots, x_i and the single variable y . The CCA approach examines this relationship over a large number of inputs (x_1, x_2, \dots, x_i) and outputs (y_1, y_2, \dots, y_j), which is performed with the help of parameters consisting of the most suitable coefficients that would maximize the correlation between these two multivariate datasets [30].

Canonical variables are parameters that optimize the correlation between these two datasets and define the best linear relationship according to each other. The coefficients that enable this linear and optimal relationship to be established according to the input data are defined as the canonical weights. The relationship between input datasets, canonical weights (b_i, a_i), and canonical variables (V_i, U_i) is given in Equations (1) and (2).

$$V_1 = b_1x_1 + b_2x_2 + b_3x_3 + \dots + b_ix_i \quad (1)$$

$$U_1 = a_1y_1 + a_2y_2 + a_3y_3 + \dots + a_iy_i \quad (2)$$

In the CCA approach, canonical variables are obtained depending on the size of the dataset, and the canonical correlation coefficient (R) values are obtained in relation to these variables. The eigenvalue problem in Equations (3) and (4) needs to be solved to obtain the canonical variables that are important for canonical correlation and to provide a linear relationship between the datasets. X and Y represent the datasets which are used in canonical correlation analysis, and C_{xy} is the covariance between them.

$$R = \frac{E[U_x V_y]}{E[U_x^2]E[V_y^2]} \quad (3)$$

$$R = \frac{E[a^T X Y^T b]}{\sqrt{E[a^T X Y^T a]E[b^T Y Y^T b]}} = \frac{a^T C_{xy} b}{\sqrt{a^T C_{xx} a b^T a^T C_{yy} b}} \quad (4)$$

A schematic representation of CCA with canonical variables, coefficients, and other parameters is shown in Figure 2 [31].

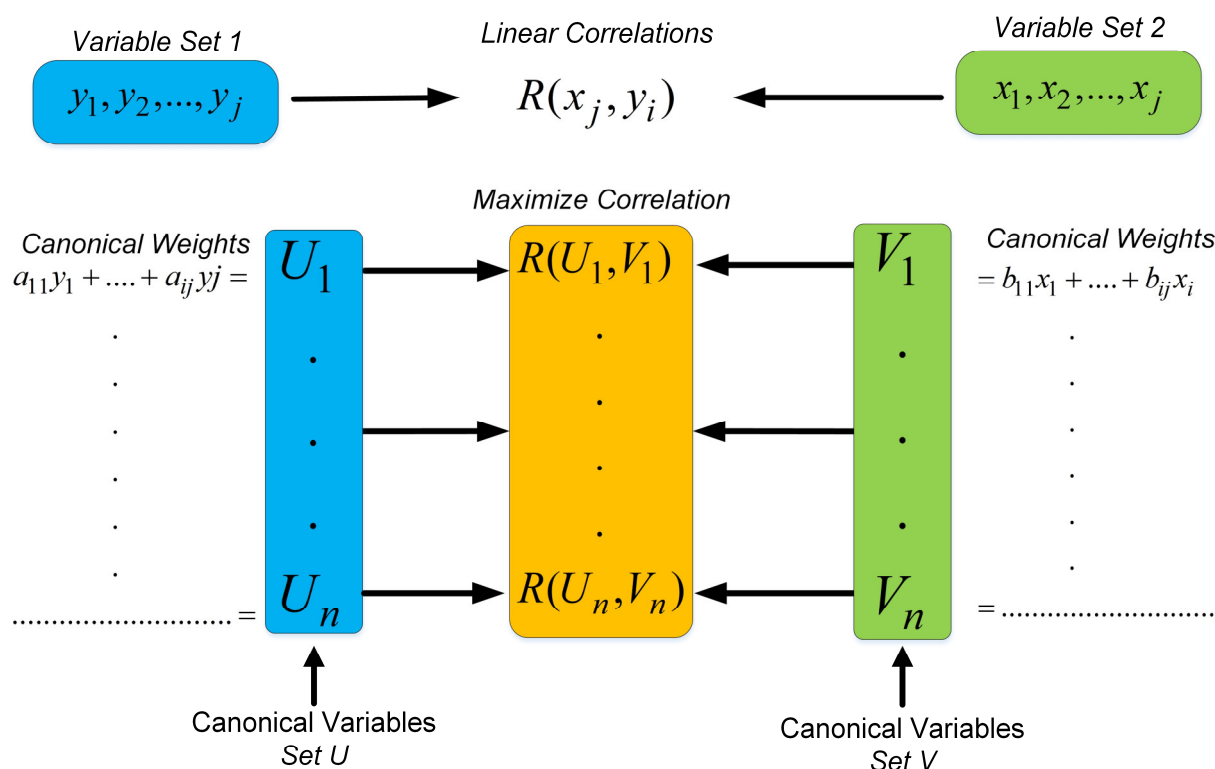


Figure 2. Schematic diagram of CCA with parameters.

Pairs of canonical variables were obtained from a high to a low correlation coefficient value because of the CCA process. Various parameters were used to test the significance of canonical variable pairs obtained from the analysis. The first is the correlation coefficient, followed by Wilks' lambda test. Wilks' lambda test, which is computed over the correlation coefficient as given in Equation (5), provides preliminary information about the obtained canonical correlation coefficients and canonical variables, as well as preliminary information about their importance. Moreover, to examine the significance of the derived canonical variables, various test methods like F-static, degrees of freedom, and chi-square (X^2) tests are available [32].

$$\Lambda = \prod (1 - R_i^2) \quad (5)$$

While determining the variables affecting cognitive load in the programming process, the CCA approach is preferred for various reasons. CCA, which includes more than one dependent variable, goes beyond the multivariate regression analysis used when there is only one dependent variable. The advantages of CCA over traditional correlation analysis originate from its being based on a specific coordinate system. The CCA has the analytical solution in closed form. Moreover, CCA is invariant to scaling. Nonlinear methods require predefined kernel functions, and the correct choice of this kernel function and its related parameters directly affect the analysis's success. Also, in nonlinear approaches, it is challenging to project back from the kernel space to the original feature space, which leads to difficulties in interpreting and visualizing the outputs of nonlinear approaches. In this context, the CCA was preferred because statistically significant and sufficient results were obtained.

2.2. Cognitive Load and EMIP Dataset

While analyzing the cognitive loads of programmers during programming processes, the EMIP dataset was used [33]. The EMIP dataset contains 216 individuals with different experience levels, genders, languages, and ages. It includes the eye movements of

programmers during the process of two code comprehension tasks, and this comprehensive dataset is publicly available. The SMI RED250 eye-tracking device was used to create the EMIP dataset. The related eye tracking device has a sampling rate of 250 Hz and an accuracy value of $<0.4^\circ$, and thanks to the high sample rate, it is also suitable for saccadic eye movements detection. When creating the EMIP dataset, two multiple-choice code comprehension tasks were presented to programmers [34]. These tasks are called the Rectangle and Vehicle tasks, respectively. The Rectangle task involves creating a class with four coordinate variables, a constructor, and methods for calculating the area, width, and height. The Vehicle task requires the development of a vehicle class with a set of variables, a constructor, and an acceleration function that can set the current speed variable. The primary method initiates and updates the development of a single object. Programmers must correctly determine the code functions in these operations using a set of multiple-choice options. Descriptive statistics of the characteristics of programmers for the EMIP dataset are presented in Table 1 [35].

Table 1. Descriptive statistics for the EMIP dataset.

Parameters	Value
Number of Males	175
Number of Females	41
Mean (Age)	26.560
Standard Deviation (Age)	9.276
Mean (Time of Experience in Programming–Year)	2.259
SD (Time of Experience in Programming–Year)	3.342

The variables listed in Table 2 were used to create a meaningful relationship between the personal parameters of computer programmers and their cognitive load. According to related studies, there is a significant relationship between cognitive load and eye movements [36]. When creating the relevant dataset, parameters that model eye movement according to time were preferred as dependent variables [26,28]. The personal parameters of the computer programmers who created the EMIP dataset were used as the independent variables. In parallel with the literature, the dependent variables used in the related study consisted of meaningful attributes extracted from the eye movements of the programmers and the time to perform the given tasks. While creating a set of dependent and independent variables, we created a structure that provided a meaningful cause-and-effect relationship. Peak rate of eye movement, a dependent variable, represents rapid eye movements and is extracted from eye movements using adaptive signal-processing techniques. The other dependent variable used in this study was change in the position and radius of the pupil over time. The last dependent variable used in the study was the time to complete the related specific tasks of the programmers.

Table 2. Independent and dependent variables.

Independent Variables	Dependent Variables
1. Age	1. Total Duration
2. Experience Period in Experiment Language	2. Standard Deviation of Pupil Radius change
3. Linguistic Distance	3. Peak Ratios of Eye Movements
4. English Level	4. Standard Deviation of Eye Movement
5. Expertise Level in Programming	
6. Frequency in Experiment Language	
7. Time of Experience in Programming	
8. Expertise Level in Experiment Language	
9. Frequency in Other Languages	
10. Gender	

2.3. Linguistic Distance

In the EMIP dataset, some data are categorically defined, and transformation is required for multivariate analysis operations, such as the native language of programmers. Linguistic distance is the measure to which one language or dialect differs from another and is used for this transformation in this study [37]. To achieve this transformation, the results of a study involving many participants in the field of linguistics were used. The related research focuses on the challenges Americans face when learning foreign languages and aims to develop a quantitative measure of the distance between English and other languages. The transformation of linguistic distance from categorical to quantitative data is presented in Table 3. It shows the linguistic distance scores of sample languages from the EMIP dataset. According to the table, Chinese is the most distant, and Italian is the closest to English. The linguistic distance was expected to be important for this study because code comprehension tests and programming language command sets were created in English when developing the EMIP dataset.

Table 3. Linguistic distance of different languages from English [37].

Categorical Data (Language)	Quantitative Data (Language Score)
German	2.25
Italian	2.5
Portuguese	2.5
Spanish	2.25
Finnish	2.00
Turkish	2.00
Greek	1.75
Thai	2.00
Chinese	1.50

2.4. Code Comprehension Tasks

Two code comprehension tasks were used to create the EMIP dataset. In the first task, the code related to the rectangular area was given, and the other was a task related to vehicle speed. Programmers are expected to look at the code and correctly determine its function among multiple options. The EMIP dataset was created by recording the eye movements of the programmers at these stages. Java code and multiple options for these two tasks are shown in Figures 3 and 4, respectively [33].

```

public class Rectangle {
    private int x1, y1, x2, y2;
    public Rectangle ( int x1, int y1, int x2, int y2) {
        this.x1=x1;
        this.y1=y1;
        this.x2=x2;
        this.y2=y2;
    }

    public int width() {return this.x2-this.x1;}
    public int height() {return this.y2-this.y1;}
    public double area() {return this.width()*this.height();}
    public static void main() (String[] args) {
        Rectangle rec1=new Rectangle ( 0, 0, 10, 10 );
        System.out.println(rect1.area());
        Rectangle rec1=new Rectangle ( 5, 5, 10, 10 );
        System.out.println(rect2.area());
    }
}

```

The program:

- computes the area of rectangles by multiplying their width (x1-x2) and height (y1-y2).
- computes the area of rectangles by multiplying their width (x2-x1) and height (y2-y1).
- computes the area of rectangles by multiplying their width (x1-y1) and height (x2-y2).
- I'm not sure.

(a)
(b)

Figure 3. Rectangle code comprehension task. (a) Java code. (b) Multiple choice for Rectangle.


```

public class Vehicle {
    String producer , type ;
    int topSpeed, currentSpeed;

    public Vehicle ( String p , String t , int tp ) {
        this.producer = p;
        this.type = t;
        this.topSpeed =tp;
        this.currentSpeed = 0 ;
    }

    public int accelerate (int kmh ) {
        if( ( this.currentSpeed + kmh ) > this.topSpeed ) {
            this.currentSpeed =this.topSpeed ;
        } else {
            this.currentSpeed = this.currentSpeed+kmh;
        }
        return this.currentSpeed ;
    }
    public static void main (String args [] ) {
        Vehicle v=new Vehicle ("Audi" , " A6" , 200 ) ;
        v.accelerate ( 10 );
    }
}

```

(a)

The program:

- defines a vehicle by producer that has a type and can reduce its speed.
- defines a vehicle by producer that has a type and can accelerate its speed.
- defines a vehicle by producer that has a type and can accelerate and reduce its speed.
- I'm not sure.

(b)

Figure 4. Vehicle class code comprehension task. (a) Java code for vehicle class. (b) Multiple choice for vehicle class.

2.5. Eye Movement Analysis

Eye movements are recorded with a sampling rate of 250 Hz, which is sufficient for detecting rapid eye movements. Saccadic eye movements are the leading eye movements associated with cognitive load. Saccadic eye movement is a rapid, conjugate eye movement in which the focus of the gaze moves from one area of the visual field to another. Rapid eye movements must be detected to determine saccadic eye movement. Therefore, changes in pupil coordinates were used in this study.

To determine saccadic eye movements from coordinate changes, a z-score-based peak detection method is preferred. The z-score concept is generally preferred in the normalization process of data, and its formula is given in Equation (6) [38].

$$z = \frac{x - \bar{x}}{\sigma_x} \quad (6)$$

where \bar{x} is the mean of the vector and σ_x is the standard deviation of the vector. In the z-score-based peak detection algorithm, the influence of previous data on the vector is meaningful and valuable. Therefore, z-score-based peak detection can be used in various applications, such as physiological signal processing, sound processing, genetics, and fluid dynamics. Equations (7)–(9) are related to the z-score-based peak detection process. The mean and standard deviation (SD) of the specific part of the signal were calculated using Equation (7). The related approach is a peak detection approach that requires effective and meaningful thresholding [39].

$$\bar{s}_i = \frac{1}{l} \sum_{i=1}^{i+l} s_i \quad \sigma_{s_i} = \sqrt{\frac{\sum_{i=1}^{i+l} (s_i - \bar{s}_i)^2}{l-1}} \quad (7)$$

$$s_i = i_n x_i + (1 - i_n) \quad s_{i-1} z_i = \frac{s_i - \bar{s}_{i-1}}{\sigma_{s_{i-1}}} \quad (8)$$

$$th_s = constant \quad y_i = \begin{cases} 1 & \text{if } |z_i| \geq th_s \\ 0 & \text{if } |z_i| < th_s \end{cases} \quad (9)$$

where th_s is the threshold value, l is the lag value, and i_n is the influence factor, which can vary between 0 and 1. It defines the effect between the previous values of the signal and the new values. A low threshold allows noise in the signal to be identified as peaks, whereas a high threshold makes it difficult to identify true peaks. The lag value defines the size of the observation screen used for the peak detection analysis. A high lag reduces the sensitivity of the algorithm, whereas a low lag causes noise in the signal to be determined as the peak value. In this study, the parameters of the z-score-based peak detection algorithm for the detection of saccadic eye movements from pupil coordinates have been selected as follows: $th_s = 5$, $l = 5$, and $i_n = 0.5$. These parameters were determined experimentally to optimize the signal-to-noise ratio and to prevent noise signals from being detected as peaks. An example output plot of the peaks obtained by the z-based peak detection method from eye movements in the EMIP dataset is shown in Figure 5.

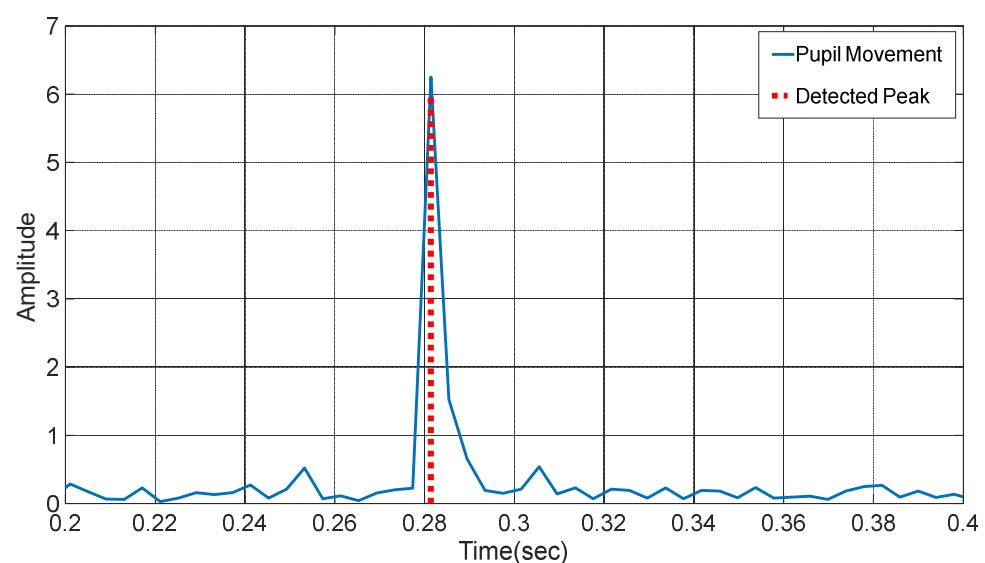


Figure 5. Movement of pupil and detected saccadic movement via z-score-based peak detection.

Figure 6 shows the changes in the pupil diameter and position over time. While the outer section of the generated circles represents the pupil diameter, the circle in the middle represents the pupil center coordinate. To reduce the noise in the raw measurement signals of eye movements and pupil diameters, Savitzky–Golay filtering, a polynomial fitting-based finite impulse response (FIR) smoothing filter, was used for pre-processing. To apply the Savitzky–Golay filter, the frame size and degree of the polynomial to be fitted must be defined, and a low-order polynomial was fitted through the data points using the least squares method [40]. In general, Savitzky–Golay filters improve the SNR ratios more than standard averaging FIR filters. Therefore, it is often preferred to reduce the noise ratio in physiological signals such as EEG, EKG, and EMG in the literature [41]. Eye movement data were made available after these signal-processing operations to build a meaningful correlation.

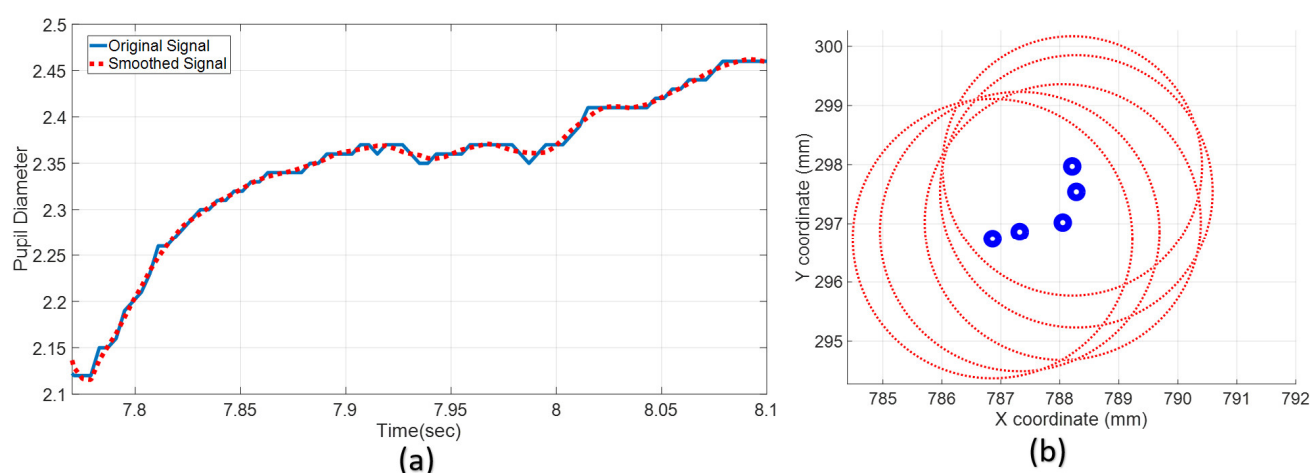


Figure 6. Pupil diameter and center coordinate. (a) Diameter change. (b) Coordinate change.

3. Results

The canonical correlation analysis method was used to establish a meaningful and interpretable relationship between eye parameters affecting cognitive load and personal parameters of programmers. In the first stage, the data were normalized using the z-score normalization method as a pre-processing step. Partial correlation analysis was performed with the dependent and independent variables created within the scope of this study, according to Table 2, and the results are given in Tables 4 and 5, respectively. The aim of the particle correlation analysis, which is performed as a preliminary analysis stage, is to reveal a meaningful correlation between and within variables [42].

Table 4. Particle correlation analysis results for dependent variables.

Variables	1	2	3	4
1	1.000	0.003	−0.026	−0.013
2	0.003	1.000	−0.048	0.244
3	−0.026	−0.048	1.000	0.009
4	−0.013	0.244	0.009	1.000

As seen in the partial correlation results for the dependent variables, there is no strong relationship between these variables. Among the dependent variables, there is a weak relationship between eye movement SD and pupil radius change SD ($r_{42} = 0.244$).

Table 5. Particle correlation analysis results for independent variables.

Variables	1	2	3	4	5	6	7	8	9	10
1	1.000	0.051	0.148	−0.065	−0.230	−0.129	0.660	0.033	0.021	−0.144
2	0.051	1.000	−0.089	0.051	−0.203	−0.114	0.379	0.417	0.076	−0.032
3	0.148	−0.089	1.000	0.399	0.001	0.045	−0.105	0.054	−0.096	−0.099
4	−0.065	0.051	0.399	1.000	−0.027	−0.018	0.072	0.022	0.057	−0.100
5	−0.230	−0.203	0.001	−0.027	1.000	−0.353	0.447	0.613	0.397	0.055
6	−0.129	−0.114	0.045	−0.018	−0.353	1.000	0.027	0.600	0.062	0.155
7	0.660	0.379	−0.105	0.072	0.447	0.027	1.000	−0.139	−0.120	0.106
8	0.033	0.417	0.054	0.022	0.613	0.600	−0.139	1.000	−0.045	0.019
9	0.021	0.076	−0.096	0.057	0.397	0.062	−0.120	−0.045	1.000	0.091
10	−0.144	−0.032	−0.099	−0.100	0.055	0.155	0.106	0.019	0.091	1.000

The highest correlations between independent variables, which are personal parameters of programmers, are calculated follows: between the age of programmer and time of experience in programming ($r_{71} = 0.66$) and between expertise level in experiment language and expertise level in programming ($r_{85} = 0.613$).

The correlation between independent and dependent variables was analyzed to establish a significant correlation between cognitive load and the personal parameters of programmers, and the results are shown in Table 6.

Table 6. Particle correlation analysis results between independent and dependent variables.

	Variables	Dependent Variables			
		1	2	3	4
Independent Variables	1	0.300	0.318	−0.141	0.182
	2	0.404	0.183	−0.003	0.011
	3	−0.059	0.097	−0.156	0.159
	4	−0.078	−0.006	0.083	0.056
	5	0.111	0.058	0.060	−0.062
	6	−0.083	0.097	0.219	−0.136
	7	0.330	0.235	−0.051	0.091
	8	0.137	0.074	0.054	−0.084
	9	0.025	−0.009	0.086	−0.085
	10	0.039	−0.104	0.033	−0.052

According to the results of the analysis, the dependent and independent variable groups with the highest correlation values are shown in Table 7 with p -values. As seen in the table, the relationship between these dependent and independent variables is statistically significant.

Table 7. Particle correlation analysis results for dependent variables.

Independent Variables	Dependent Variables	Correlation Coefficient
Experience Period in Experiment Language	Total Duration	0.404 *
Time of Experience in Programming	Total Duration	0.330 *
Age	SD of Pupil Radius Change	0.318 *

Note: * $p < 0.001$

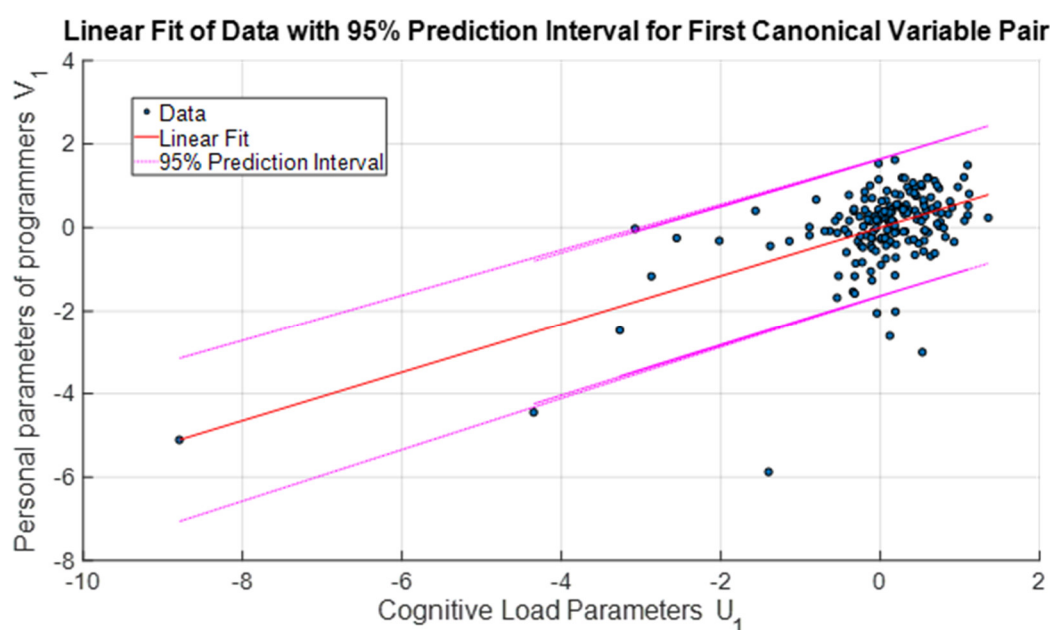
As shown by the results of the partial correlation analysis between the dependent and independent variables, there was a moderate correlation between the time it took programmers to complete the given tasks and their experience in the experimental language. In this study, moderate or weak correlation was determined according to the rule of thumb used to interpret the magnitude of the correlation coefficient [43]. A multivariate data analysis approach is required to improve the correlation coefficient and establish a more effective and meaningful relationship between the two sets of variables. For this reason, canonical correlation analysis was preferred, because it analyzes the effects on the variables in a meaningful manner and provides the parameters that will provide the maximum correlation between the two datasets. The correlation ratio between the datasets and the level of significance of the variables was examined using canonical correlation analysis. Canonical correlation analysis provided the canonical correlation coefficient between cognitive load parameters and programmers' personal parameters, as well as canonical variable vectors and weights. The relationship between the two datasets was analyzed to determine the significance of the canonical correlation coefficients. The canonical correlation coefficients for the canonical variable vectors among the relevant datasets are listed in Table 8.

Table 8. Correlation coefficient between canonical variables vectors.

Number of Canonical Variables	Canonical Correlation Coefficient
1	0.581 *
2	0.344
3	0.313
4	0.103

Note: * $p < 0.001$

When the particle correlation analysis and canonical correlation results of the relationship between cognitive load and the databases of programmers' parameters are analyzed, it is seen that canonical correlation analysis yields more statistically significant results, as expected. In this way, the relationship between the dependent and independent variables was found statistically more meaningfully and became suitable for integrated analysis. Figures 7 and 8 show the variation in the 1st and 2nd canonical variable vectors representing these relationships, respectively. When the related figures were analyzed, it was determined that the relationship between the first canonical variable pair was more linear and statistically more significant. Therefore, the first canonical variable pair was preferred for creating a mathematical model to define the relationship between programmers' cognitive load and personal parameters.

**Figure 7.** First pair (U_1 and V_1).

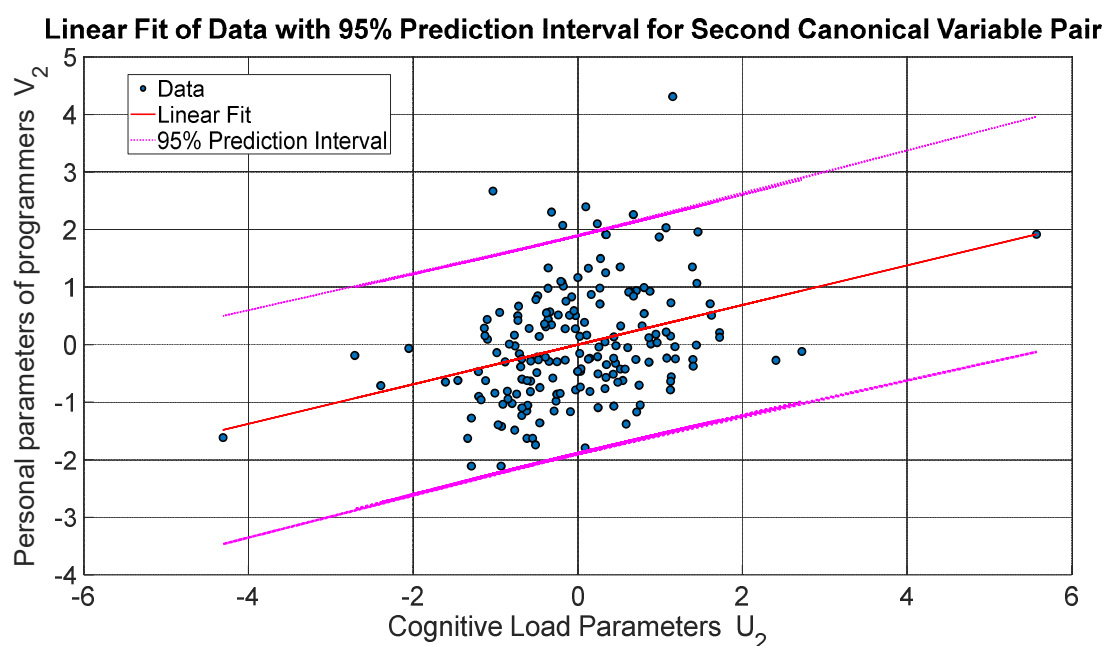


Figure 8. Second pair (U_2 and V_2).

When the relationship between the two canonical variable vector pairs is examined, it is clearly observed that the relationship of the first vector pair has the highest correlation and a more linear characteristic. In addition, this linear characteristic difference can be easily observed from the deviation in the distribution of data on the fitted optimal line. Using canonical weighting, which is statistically significant and has the highest correlation coefficient, a path diagram describing the relationship between cognitive load and the personal parameters of programmers was generated, as shown in Figure 9.

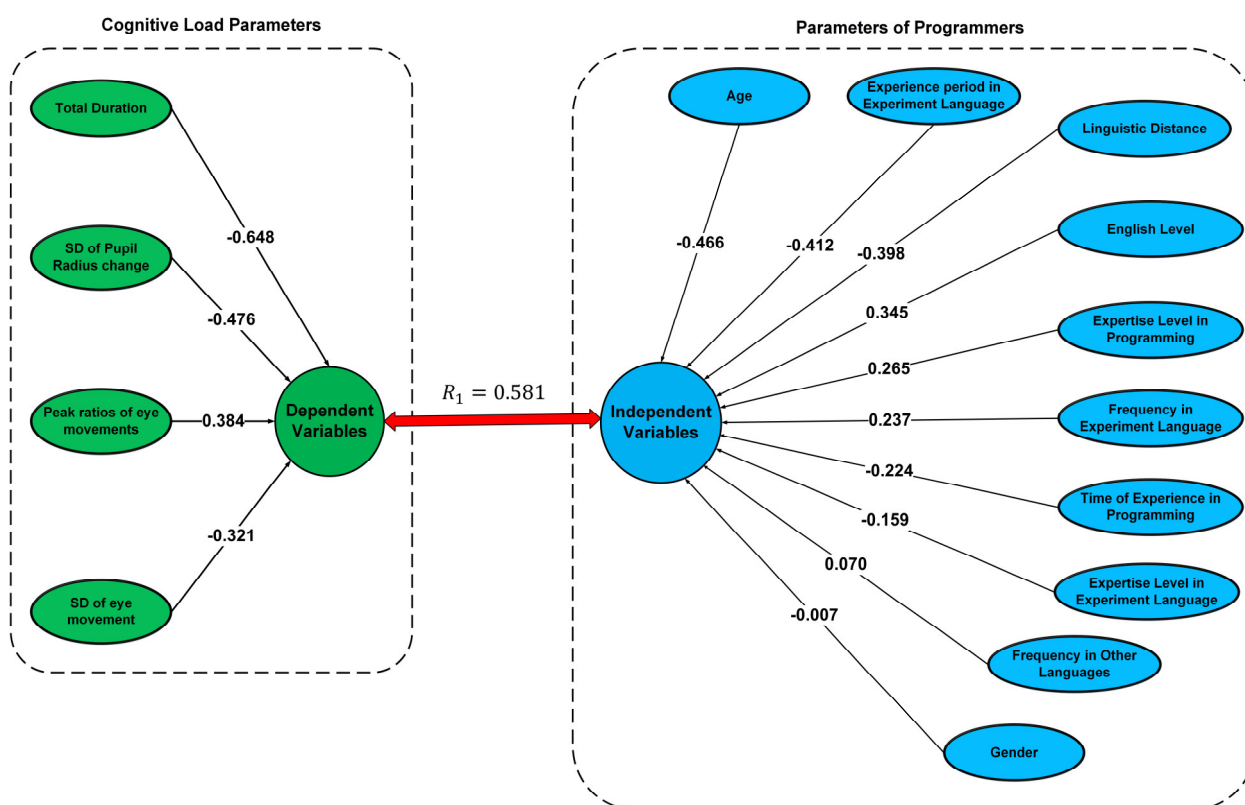


Figure 9. Path diagram of cognitive load and personal parameters.

The amplitude of the canonical weights of canonical variables provides information regarding the influence of variables according to the datasets. Therefore, the relatively high amplitude of the weights provides information regarding the effect ratio between each dataset. To make the path diagram more meaningful using canonical weights, the percentage effects of the normalized canonical weights on the variables are shown in Figure 10.

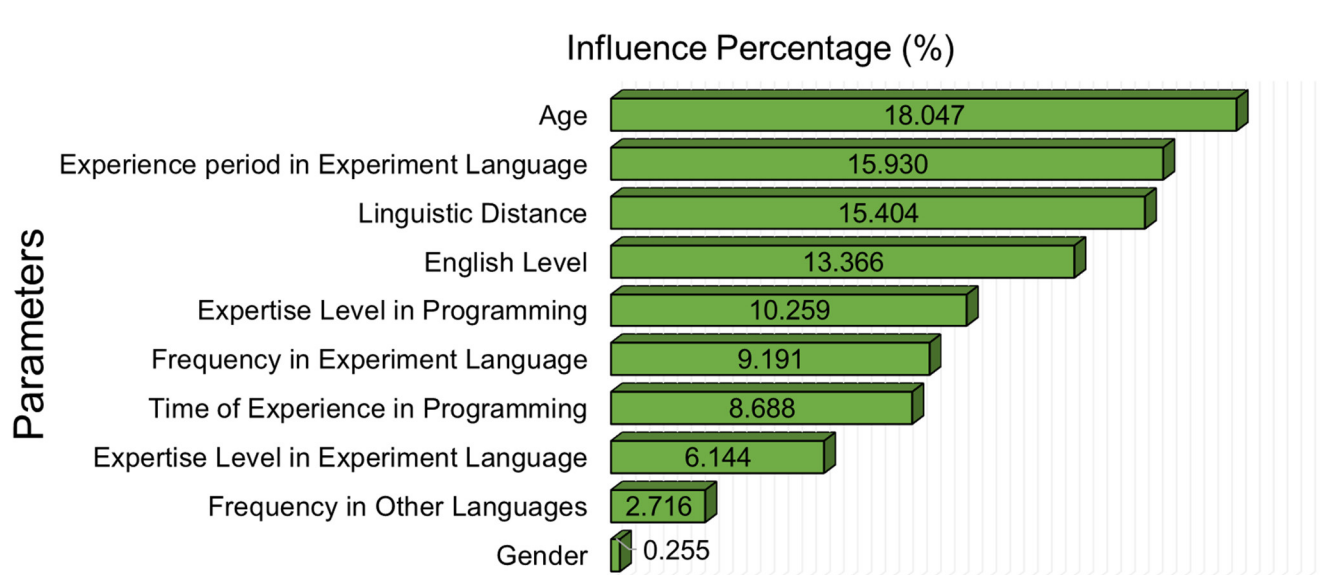


Figure 10. Factors influencing the cognitive load of programmers.

When the percentages of personal parameters affecting the cognitive load parameters of programmers were examined, it was determined that the highest effect was that of the age factor (18.047%), and the gender parameter had the lowest effect (0.255%). One of the important and novel results of this study is that the linguistic distance between a programmer's native language and English has a significant effect on cognitive load (15.404%). When the canonical correlation analysis of the dataset is examined from another point of view, it is seen that there is a more balanced weight between the cognitive load parameters, which are as follows: peak ratio (20.9%), SD of eye movement (17.544%), SD of pupil radius (26.019%), and total duration (35.436%).

4. Discussion

In this study, we have determined the parameters that affect the cognitive load of computer programmers during code comprehension tasks in percentage terms. We have used the EMIP database, a comprehensive database of eye movements and different personnel parameters of 216 participants [33,34].

We have transformed categorical variables, such as the native language of programmers in the relevant EMIP database, into meaningful and efficient quantitative variables using the linguistic distance approach [37]. We have extensively investigated the influence of linguistic distance on the cognitive load of computer programmers in this study for the first time in the literature with integrated sub-parameters.

To determine the rapid movements of the eye, the time-dependent changes in the pupil coordinates have been obtained, and the peak values for modelling the sudden movement of the eye have been determined with the z-score peak detection algorithm. We have used the Savitzky–Golay FIR smoothing filter to reduce signal measurement noise during data processing. We have assessed eye movements and problem-solving time as cognitive load parameters of programmers. The personal parameters of the programmers and the calculated cognitive load parameters have been determined as dependent and independent variables. First, we performed partial correlation analyses within and between these variables and tested the statistical significance of the results obtained. In the

next step, we analyzed the effect of these variables on each other with the CCA approach to increase statistical significance and reveal more comprehensive results. As a result of the analysis, we have observed that not only the native language spoken but also the linguistic distance of the native language from English significantly affected the cognitive load. The results that we have obtained within the scope of this study support the results and suggestions of the studies examining the relationship between cognitive load and language [44].

We have determined the relationship between computer programming and spoken languages is essential according to cognitive load theory. Therefore, like the continuous exposure method used in foreign language education, teaching programming languages like spoken language can be meaningful and beneficial [45]. We have hypothesized that this approach could positively influence the process of learning a programming language in the future.

We found that the age of the programmer had the highest effect on the cognitive load. Then, we determined that the duration of experience in the relevant programming language also had a significant effect on the programmer's cognitive load. These outputs support the studies finding that age is an essential factor in cognitive load [46]. Studies conducted within the scope of analyzing learning processes have revealed that cognitive load increases with age [47].

One of the results of this study is that we have found that the frequency of using a different programming language had less effect on cognitive load. We have observed that specialization in or intensive use of a programming language with a different structure has a limited effect on the perception and interpretation of the problem encountered in another programming language. However, to support and generalize this inference, conducting comprehensive and specific studies that analyze the effect of different programming languages on cognitive load will be helpful.

It is widely recognized that there are biological and social differences between the ways men and women perceive environmental stimuli [48]. However, current theories on cognitive load argue against the idea that these differences lead to precise and direct differences in working memory cognitive load. Cognitive abilities and performance are influenced by not only gender but also past experiences, education, and other factors that account for individual cognitive differences [49].

A comprehensive review study conducted in this context examined many neuroimaging studies to investigate gender differences in brain structures. Although researchers have found gender-related differences in some brain regions, these differences are insufficient to explain cognitive abilities [50].

Another comprehensive meta-analysis study investigating visual-spatial working memory in terms of gender differences has similar findings. The meta-analysis examined gender differences in visual-spatial working memory using a dataset of 180 effect sizes from 98 samples of healthy men and women aged 3 to 86 years. Statistical analyses of the study showed that these differences were too small to adequately explain gender differences in visual-spatial working memory abilities, particularly in mental rotation, where effect sizes may be relatively large [51].

As a result of this study, we have observed that gender difference has a limited effect on cognitive load in computer programming processes. One point not entirely comprehended in research on gender differences in cognitive load has been the deficiency of clear universal spatial or temporal patterns in males' and females' responses to external factors. In this study on computer programming and cognitive load, similar results to those in the literature have been obtained, and no significant correlation has been found between cognitive load and gender differences. The pattern of the relationship between gender differences and cognitive load will become more evident and meaningful with comprehensive studies with different stimuli over time.

This study has some limitations that should be emphasized. The programmers' cognitive load or physiological fatigue levels were not measured physiologically or with

various scales before performing the tasks. In the process of realizing the dataset, the programmers only looked at two code samples; both were object-oriented, so whether the obtained findings can be generalized to more algorithmic code or code in languages with different programming models may need to be clarified. As this was a multi-location study, minor differences in the experimental setup may have occurred despite using the same hardware in all locations, such as the same eye tracker and laptop. The use of eye tracking to measure cognitive load has important advantages, but in some cases, it can also have disadvantages. Studies conducted with eye-tracking systems can sometimes feel uncomfortable or artificial for participants. Therefore, they may cause changes in participant behavior that may affect the results of the studies.

5. Conclusions

The influence of personal parameters on the cognitive load in computer programming processes was examined in detail in this study. In this innovative study, the effect of code structures, which form the basis of computer programming, on cognitive load was investigated using the linguistic distance metric. The EMIP dataset, one of the most comprehensive datasets in this field, consisting of 216 programmers, was used in this study. Eye movements in two different code comprehension tasks and various personal parameters of the participants are available in the relevant datasets. In this study, the z-score-based peak estimation algorithm was used to determine rapid saccadic eye movement signals, an essential indicator of cognitive load. At the same time, the change in pupil diameter, another indicator of cognitive load change, was also analyzed. In the analysis, the statistical interpretation and validity of the results were tested in detail. The relationship between these comprehensive datasets was examined using canonical correlation analysis. The canonical weights obtained from the analysis were normalized, and the percentage effects of the parameters were calculated relative to each other. It was determined that the effect of linguistic distance analyzed within the scope of this study on cognitive load in the computer programming process is significantly important. It has been observed that the frequency of using a programming language other than that used in code comprehension tasks has a weak effect on cognitive load. Considering this result, the use of an alternate programming language has little effect on the cognitive load of programming in another language. It was determined that the effect of gender on cognitive load in the computer programming process was weak. In this regard, the cognitive load in computer programming processes can be independent of gender. It has been determined that the English level of programmers significantly impacts cognitive load in computer programming.

Author Contributions: Conceptualization, D.I.; Data curation, D.I.; Formal analysis, D.I.; Investigation, D.I., M.C.C. and F.D.; Methodology, D.I.; Resources, M.C.C.; Software, D.I.; Supervision, F.D.; Validation, D.I. and M.C.C.; Visualization, D.I. and M.C.C.; Writing—original draft, D.I.; Writing—review and editing, D.I., M.C.C. and F.D. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data presented in this study are available on reasonable request from the corresponding author.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Aagten-Murphy, D.; Bays, P.M. Functions of memory across saccadic eye movements. In *Processes of Visuospatial Attention and Working Memory*; Springer International Publishing: Cham, Switzerland, 2018; pp. 155–183.
2. JothiPrabha, A.; Bhargavi, R.; Rani, B.D. Prediction of dyslexia severity levels from fixation and saccadic eye movement using machine learning. *Biomed. Signal Process. Control.* **2023**, *79*, 104094.

3. Gunawardane PD, S.H.; MacNeil, R.R.; Zhao, L.; Enns, J.T.; de Silva, C.W.; Chiao, M. A fusion algorithm for saccade eye movement enhancement with EOG and lumped-element models. *IEEE Trans. Biomed. Eng.* **2021**, *68*, 3048–3058.
4. Pfleging, B.; Fekety, D.K.; Schmidt, A.; Kun, A.L. A model relating pupil diameter to mental workload and lighting conditions. In Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems, San Jose, CA, USA, 7–12 May 2016; pp. 5776–5788.
5. Kun, A.L.; Palinko, O.; Medenica, Z.; Heeman, P.A. On the feasibility of using pupil diameter to estimate cognitive load changes for in-vehicle spoken dialogues. In Proceedings of the 14th Annual Conference of the International Speech Communication, Lyon, France, 25–29 August 2013; pp. 3766–3770.
6. Kruger, J.L.; Hefer, E.; Matthew, G. Measuring the impact of subtitles on cognitive load: Eye tracking and dynamic audiovisual texts. In Proceedings of the 2013 Conference on Eye Tracking South Africa, Cape Town, South Africa, 9 August 2013; pp. 62–66.
7. Chen, S.; Epps, J. Automatic classification of eye activity for cognitive load measurement with emotion interference. *Comput. Methods Programs Biomed.* **2013**, *110*, 111–124.
8. Zagermann, J.; Pfeil, U.; Reiterer, H. Measuring cognitive load using eye tracking technology in visual computing. In Proceedings of the Sixth Workshop on Beyond Time and Errors on Novel Evaluation Methods for Visualization, Baltimore, MD, USA, 24 October 2106; pp. 78–85.
9. Cong, R.; Tago, K.; Jin, Q. Measurement and verification of cognitive load in multimedia presentation using an eye tracker. *Multimed. Tools Appl.* **2022**, *81*, 26821–26835.
10. Parbat, D.; Chakraborty, M. A Novel Methodology to study the Cognitive Load Induced EEG Complexity Changes: Chaos, Fractal and Entropy based approach. *Biomed. Signal Process. Control.* **2021**, *64*, 102277.
11. Patel, V.L.; Kaufman, D.R. Cognitive informatics. In *Biomedical Informatics*; Springer: Cham, Switzerland, 2021; pp. 121–152.
12. Sweller, J. Cognitive load theory and educational technology. *Educ. Technol. Res. Dev.* **2020**, *68*, 1–16.
13. Paas, F.; Ayres, P. Cognitive load theory: A broader view on the role of memory in learning and education. *Educ. Psychol. Rev.* **2014**, *26*, 191–195.
14. Van Merriënboer, J.J.G.; Sweller, J. Cognitive Load Theory and Complex Learning: Recent Developments and Future Directions. *Educ. Psychol. Rev.* **2005**, *17*, 147–177.
15. Klepsch, M.; Schmitz, F.; Seufert, T. Development and Validation of Two Instruments Measuring Intrinsic, Extraneous, and Germane Cognitive Load. *Front. Psychol.* **2017**, *8*, 1997.
16. De Jong, T. Cognitive load theory, educational research, and instructional design: Some food for thought. *Instr. Sci.* **2009**, *38*, 105–134.
17. Fraser, K.L.; Meguerdichian, M.J.; Haws, J.T.; Grant, V.J.; Bajaj, K.; Cheng, A. Cognitive Load Theory for debriefing simulations: Implications for faculty development. *Adv. Simul.* **2018**, *3*, 28.
18. Orru, G.; Longo, L. The evolution of cognitive load theory and the measurement of its intrinsic, extraneous and germane loads: A review. In *International Symposium on Human Mental Workload: Models and Applications*; Springer: Cham, Switzerland, 2018; pp. 23–48.
19. Mitre-Hernandez, H.; Carrillo, R.C.; Lara-Alvarez, C. Pupillary Responses for Cognitive Load Measurement to Classify Difficulty Levels in an Educational Video Game: Empirical Study. *JMIR Serious Games* **2021**, *9*, e21620.
20. Keskin, M.; Ooms, K.; Dogru, A.O.; De Maeyer, P. Exploring the Cognitive Load of Expert and Novice Map Users Using EEG and Eye Tracking. *ISPRS Int. J. Geo-Inf.* **2020**, *9*, 429.
21. Souchet, A.D.; Philippe, S.; Lourdeaux, D.; Leroy, L. Measuring Visual Fatigue and Cognitive Load via Eye Tracking while Learning with Virtual Reality Head-Mounted Displays: A Review. *Int. J. Human Comput. Interact.* **2021**, *38*, 801–824.
22. Que, Y.; Zheng, Y.; Hsiao, J.H.; Hu, X. Studying the effect of self-selected background music on reading task with eye movements. *Sci. Rep.* **2023**, *13*, 1704.
23. Krejtz, K.; Duchowski, A.T.; Niedzielska, A.; Biele, C.; Krejtz, I. Eye tracking cognitive load using pupil diameter and microsaccades with fixed gaze. *PLoS ONE* **2018**, *13*, e0203629.
24. Chatfield, C. *Introduction to Multivariate Analysis*; Routledge: England, UK, 2018.
25. Cobre, A.D.F.; Böger, B.; Villhena, R.D.O.; Fachi, M.M.; dos Santos, J.M.M.F.; Tonin, F.S. A multivariate analysis of risk factors associated with death by COVID-19 in the USA, Italy, Spain, and Germany. *J. Public Health* **2020**, *30*, 1189–1195.
26. Catalbas, M.C.; Burken, S. The mathematical relationship between COVID-19 cases and socio-economic indicators of OECD countries. *Pathog. Glob. Health* **2022**, *116*, 376–388.
27. Mihalik, A.; Chapman, J.; Adams, R.A.; Winter, N.R.; Ferreira, F.S.; Shawe-Taylor, J.; Mourão-Miranda, J.; Alzheimer’s Disease Neuroimaging Initiative. Canonical correlation analysis and partial least squares for identifying brain-behaviour associations: A tutorial and a comparative study. *Biol. Psychiatry Cogn. Neurosci. Neuroimaging* **2022**, *7*, 1055–1067.
28. Wang, H.-T.; Smallwood, J.; Mourao-Miranda, J.; Xia, C.H.; Satterthwaite, T.D.; Bassett, D.S.; Bzdok, D. Finding the needle in a high-dimensional haystack: Canonical correlation analysis for neuroscientists. *Neuroimage* **2020**, *216*, 116745.
29. Zhuang, X.; Yang, Z.; Cordes, D. A technical review of canonical correlation analysis for neuroscience applications. *Hum. Brain Mapp.* **2020**, *41*, 3807–383.
30. Hotelling, H. Relations between two sets of variates. In *Breakthroughs in Statistics: Methodology and Distribution*; Springer: New York, NY, USA, 1992; pp. 162–190.
31. Rajasundaram, D.; Runavot, J.-L.; Guo, X.; Willats, W.G.T.; Meulewaeter, F.; Selbig, J. Understanding the Relationship between Cotton Fiber Properties and Non-Cellulosic Cell Wall Polysaccharides. *PLoS ONE* **2014**, *9*, e112168.
32. Forsyth, D. Canonical Correlation Analysis. In *Applied Machine Learning*; Springer: Cham, Switzerland, 2109; pp. 139–151.

33. Bednarik, R.; Busjahn, T.; Gibaldi, A.; Ahadi, A.; Bielikova, M.; Crosby, M.; Essig, K.; Fagerholm, F.; Jbara, A.; Lister, R.; et al. EMIP: The eye movements in programming dataset. *Sci. Comput. Program.* **2020**, *198*, 102520.
34. Al Madi, N.; Guarnera, D.; Sharif, B.; Maletic, J. EMIP Toolkit: A Python Library for Customized Post-processing of the Eye Movements in Programming Dataset. In *ACM Symposium on Eye Tracking Research and Applications*; Association for Computing Machinery: New York, NY, USA, 2021; pp. 1–6. <https://doi.org/10.1145/3448018.3457425>.
35. Eye Movements in Programming Dataset. Available online: <http://www.emipws.org/> (accessed on 1 February 2023).
36. Berssanette, J.H.; de Francisco, A.C. Cognitive Load Theory in the Context of Teaching and Learning Computer Programming: A Systematic Literature Review. *IEEE Trans. Educ.* **2021**, *65*, 440–449.
37. Chiswick, B.R.; Miller, P.W. Linguistic Distance: A Quantitative Measure of the Distance Between English and Other Languages. *J. Multiling. Multicult. Dev.* **2005**, *26*, 1–11.
38. Jain, S.; Shukla, S.; Wadhvani, R. Dynamic selection of normalization techniques using data complexity measures. *Expert Syst. Appl.* **2018**, *106*, 252–262.
39. Catalbas, M.C.; Cegovnik, T.; Sodnik, J.; Gulten, A. Driver fatigue detection based on saccadic eye movements. In *Proceedings of the 2017 10th International Conference on Electrical and Electronics Engineering (ELECO)*, Bursa, Turkey, 30 December 2017; pp. 913–917.
40. Krishnan, S.R.; Seelamantula, C.S. On the Selection of Optimum Savitzky-Golay Filters. *IEEE Trans. Signal Process.* **2012**, *61*, 380–391.
41. Gajbhiye, P.; Mingchinda, N.; Chen, W.; Mukhopadhyay, S.C.; Wilaiprasitporn, T.; Tripathy, R.K. Wavelet Domain Optimized Savitzky–Golay Filter for the Removal of Motion Artifacts from EEG Recordings. *IEEE Trans. Instrum. Meas.* **2020**, *70*, 4002111.
42. Peng, J.; Wang, P.; Zhou, N.; Zhu, J. Partial Correlation Estimation by Joint Sparse Regression Models. *J. Am. Stat. Assoc.* **2009**, *104*, 735–746.
43. Krehbiel, T.C. Correlation coefficient rule of thumb. *Decis. Sci. J. Innov. Educ.* **2004**, *2*, 97–100.
44. Bers, M.U. Coding as another language: A pedagogical approach for teaching computer science in early childhood. *J. Comput. Educ.* **2019**, *6*, 499–528.
45. Fedorenko, E.; Ivanova, A.; Dhamala, R.; Bers, M.U. The language of programming: A cognitive perspective. *Trends Cogn. Sci.* **2019**, *23*, 525–528.
46. Van Gerven, P.W.; Paas, F.G.; Van Merriënboer, J.J.; Schmidt, H.G. Cognitive load theory and aging: Effects of worked examples on training efficiency. *Learn. Instr.* **2002**, *12*, 87–105.
47. Korbach, A.; Brünken, R.; Park, B. Differentiating different types of cognitive load: A comparison of different measures. *Educ. Psychol. Rev.* **2018**, *30*, 503–529.
48. Bevilacqua, A. Commentary: Should Gender Differences be Included in the Evolutionary Upgrade to Cognitive Load Theory? *Educ. Psychol. Rev.* **2016**, *29*, 189–194.
49. Nori, R.; Zucchelli, M.M.; Palmiero, M.; Piccardi, L. Environmental cognitive load and spatial anxiety: What matters in navigation? *J. Environ. Psychol.* **2023**, *88*, 102032.
50. Ruigrok, A.N.; Salimi-Khorshidi, G.; Lai, M.-C.; Baron-Cohen, S.; Lombardo, M.V.; Tait, R.J.; Suckling, J. A meta-analysis of sex differences in human brain structure. *Neurosci. Biobehav. Rev.* **2013**, *39*, 34–50.
51. Voyer, D.; Voyer, S.D.; Saint-Aubin, J. Sex differences in visual-spatial working memory: A meta-analysis. *Psychon. Bull. Rev.* **2016**, *24*, 307–334.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.