Juan Salvador Toledo Rios

Development of an artificial intelligence model for the detection of attention states.

2025

Universidad Autónoma de Querétaro

Facultad de Ingeniería

Development of an artificial intelligence model for the detection of attention states.

Tesis

Que como parte de los requisitos para obtener el Grado de

Maestro en

Ciencias en Inteligencia Artificial

Presenta

Juan Salvador Toledo Rios

Dirigido por:
Dr. Marco Antonio Aceves Fernández

Querétaro, Qro. a Mayo 2025

Universidad Autónoma de Querétaro

Facultad de ingeniería

Maestría en Ciencias en Inteligencia Artificial

Development of an artificial intelligence model for the detection of attention states.

Tesis

Que como parte de los requisitos para obtener el Grado de

Maestro en Ciencias en Inteligencia Artificial

Presenta

Juan Salvador Toledo Rios

Dirigido por

Dr. Marco Antonio Aceves Fernádnez

Dr. Marco Antonio Aceves Fernández
Presidente

Dr. Saúl Tovar Arriaga
Secretario

Dr. Jesús Carlos Pedraza Ortega
Vocal

Dra. María del Carmen Cabrera Hernández
Suplente

Dr. Juan Manuel Ramos Arreguín
Suplente

Centro Universitario, Querétaro, Qro. México
Fecha de aprobación por el Consejo Universitario (mes y año)

Somnium, Crede et Fac fieri

I dedicate this work to every person who believed in me and supported me, and above all I dedicate this work to my family.

Dedico este trabajo a todas las personas que creyeron en mí, me apoyaron y, sobre todo, dedico este trabajo a mi familia.

Acknowledgment

To my parents Juan Gabriel Toledo Ibarra and Susana Georgina Rios de la Rosa, who, thanks to all your unconditional support, your love, I am achieving one more goal; I thank you for allowing me to have the privilege of studying for a graduate degree without worrying about anything, for motivating me to be a better person at every moment, for having given me everything with full hands, you are my inspiration, what I aspire one day to become. To my sister Susana Gabriela, for your unconditional love, because you are always there for me supporting me, bothering us, for being my companion of adventures all this time, just like our parents, you are someone I have always admired.

To my grandparents for being second parents to me, for always being proud of me and showing it every chance they get, for supporting me through this stage, for your advice, your talks, your teachings, and anecdotes thank you for everything. I want to give a special thanks to Sofia Gabriela, who was my inspiration to be able to develop this research, for being my engine every day, for bringing joy to my life, for waiting for my return eagerly every time I leave, thank you for the purest affection, thank you for everything.

Thanks to my thesis director, Dr. Marco Antonio Aceves Fernández, for all his teachings, his patience, his guidance throughout this research and master's degree, for sharing his knowledge in a selfless way, always seeking the best for his advisors, and for his empathy.

I would like to thank my synod, Dr. Saúl Tovar, Dr. Jesús Pedraza, Dr. Carmen Cabrera, and Dr. Juan Manuel, for their time, dedication, and valuable contributions to this project; their experience and feedback have been fundamental to enriching and nurturing this research. I would especially like to thank the Universidad Autónoma de Querétaro for being part of the program I studied, for the facilities and the staff; and I would especially like to thank SECIHTI for their financial support during these years I was studying, allowing me to focus only on that.

Agradecimientos

A mis padres Juan Gabriel Toledo Ibarra y Susana Georgina Rios de la Rosa, que, gracias a todo su apoyo incondicional, su cariño, estoy logrando cumplir una meta más; les agradezco por permitirme tener el privilegio de estudiar un posgrado sin la preocupación de nada, por motivarme a ser mejor persona en cada momento, por haberme dado todo a manos llenas, ustedes son mi inspiración, lo que aspiro algún día poder llegar a ser.

A mi hermana Susana Gabriela, por su cariño incondicional, porque siempre estás ahí para mí apoyándome, molestándonos, por ser mi compañera de aventuras todo este tiempo, al igual que nuestros padres eres alguien que siempre he admirado . A mis abuelos por ser unos segundos padres para mí, por siempre estar orgullosos de mí y demostrarlo cada que pueden, por apoyarme en esta etapa, por sus consejos, sus pláticas, sus enseñanzas y anécdotas, gracias por todo.

Quiero dar un agradecimiento especial a Sofia Gabriela, quien fue mi inspiración para poder desarrollar esta investigación, por ser mi motor cada día, por brindar alegría a mi vida, por esperar mi regreso con ansias cada vez que me voy, gracias por el cariño más puro, gracias por todo.

Gracias a mi director de tesis, el Dr. Marco Antonio Aceves Fernández, por todas sus enseñanzas, su paciencia, su guía a lo largo de esta investigación y maestría, por compartir su conocimiento de forma desinteresada buscando siempre lo mejor para sus asesorados, por su empatía.

Expreso mis agradecimientos a mi sínodo, Dr. Saúl Tovar, Dr. Jesús Pedraza, Dra. Carmen Cabrera y Dr. Juan Manuel, por su tiempo, dedicación y valiosas contribuciones a este proyecto, su experiencia y retroalimentación han sido fundamentales para enriquecer y nutrir esta investigación.

En especial quiero agradecer a la Universidad Autónoma de Querétaro por haber sido parte del programa que estudié, por las instalaciones y el personal; y quiero agradecer sobre todo a SECIHTI por haber apoyado de forma financiera durante estos años en los que estuve estudiando, permitiéndome enfocarme únicamente en eso.

Abbreviations and acronyms

Artificial Intelligence	AI
Machine Learning	ML
Deep Learning	DL
Artificial Neural Network	ANN
Electrocardiogram	ECG
Electroencephalogram	EEG
Logistic Regression	LR
True Positive	TP
True Negative	TN
False Positive	FP
False Negative	FN
Support Vector Machine	SVM
Ridge Classifier	RC
Velocity-Threshold Identification	I-VT

Resumen

La capacidad cognitiva es un proceso neurológico muy complejo y a su vez muy estudiado, sobre todo en el proceso de atención debido al alto impacto que tiene sobre la calidad de vida en las personas, sin embargo, la detección y clasificación de los niveles de atención sigue siendo un desafío debido a la escasez de datos para trabajar, así como la subjetividad de los indicadores al momento de medir los niveles de atención. El presente trabajo tiene la intención de desarrollar un modelo de Inteligencia Artificial aplicado a datos de trayectorias oculares para la clasificación de los niveles de atención. Se trabajó con una base de datos pública, usando una Red Neuronal Convolucional para extraer las características de las trayectorias, empleando los vectores de características como datos de entrada en distintos modelos de Aprendizaje Automático; además se propuso una metodología para generar y validar el aumento de datos debido al desbalanceo de clases en la base de datos. El modelo de regresión logística fue seleccionado para realizar la clasificación debido a su adaptabilidad a los datos después de las pruebas, además de ser optimizado mediante la técnica de Grid Search. Como conclusión se desarrolló y optimizó un modelo híbrido incorporando técnicas de Aprendizaje Profundo con Aprendizaje Automático, probando que un modelo puede clasificar de forma acertada alcanzando una exactitud superior al 95%, superando investigaciones previas. Este estudió demostró el potencial que se tiene al trabajar con técnicas no invasivas usando Inteligencia Artificial en la clasificación de los niveles de atención.

Keywords: Inteligencia Artificial, Redes Neuronales Convolucionales, Aprendizaje Automático, Clasificación, Atención, Aumento de Datos

Abstract

Cognitive ability is a very complex neurological process and at the same time very studied, especially in the process of attention due to the high impact it has on the quality of life in people, however, the detection and classification of attention levels remain a challenge due to the scarcity of data to work with, as well as the subjectivity of the indicators when measuring the levels of attention. The present work intends to develop an Artificial Intelligence model applied to eye trajectory data for the classification of attention levels. We worked with a public database, using a Convolutional Neural Network to extract the features of the trajectories, using the feature vectors as input data in different Machine Learning models; in addition, a methodology was proposed to generate and validate the increase of data due to the imbalance of classes in the database. The logistic regression model was selected to perform the classification due to its adaptability to the data after testing, in addition to being optimized using the Grid Search technique. In conclusion, a hybrid model was developed and optimized incorporating Deep Learning techniques with Machine Learning, proving that a model can accurately classify reaching an accuracy higher than 95%, surpassing previous research. This study demonstrated the potential of working with non-invasive techniques using Artificial Intelligence in the classification of attention levels.

Keywords: Artificial Intelligence, Neural Networks, Machine Learning, Classification, Attention, Data Augmentation

Contents

List of Tables

List of Figures

1 Introduction

The correct and timely detection of the state of attention is of utmost importance due to the impact in different areas that it has on the daily life of people with any condition, since an erroneous diagnosis can generate numerous comorbidities, decreasing the quality of life of people. Despite this, so far there are very few tools that help in this detection, being mostly very intrusive. The main purpose of this work is to develop an artificial intelligence attentional model which will help in the detection of such states using eye-tracking techniques in cognitive tests, to provide a more accurate result.

1.1 Problem description

It is no easy task to identify attentional states given the subjectivity of the matter and the absence of direct measures. Since attention is a mental process that cannot be observed firsthand, we must rely on indirect indicators to deduce its presence. These indicators take various forms, including observable behaviors like screen-gazing, self-disclosure of attentional states by individuals, physiological measurements such as heart rate, as well as imaging techniques such as EEG or functional magnetic resonance imaging (fMRI) [1].

Nevertheless, each of these indicators brings its own limitations to the table: behavior may come off as misleading, self-reporting may carry an inherent bias, physiological measurements may point to other emotions, and brain imaging can only partly represent the bigger picture. Capturing attention accurately in real-life situations is complicated due to the dynamic and multifaceted nature of the process [2].

Ailing to adequately discern attentional states assumes significance within the lives of individuals afflicted with Pervasive Developmental Disorder (PDD), particularly when concomitant with Asperger's Syndrome (AS) and Attention Deficit Hyperactivity Disorder (ADHD). This significance arises due to the multifaceted nature of their conditions, encompassing challenges in social development, as well as enduring difficulties within their occupational and educational settings throughout their lifespan [3].

The major problem to be addressed in this study is to develop an artificial intelligence model that helps to validate the test to detect the attentional states through saccadic eyes movement due must of the classification are attentive, non-attentive, may have disadvantages previously described over more complete classifications.

1.2 Justification

The states of attention have been related to cognitive ability by the different states that have been proposed over time, such as sustained attention, divided attention, selective

attention and alternating attention, since many neurological disorders are often related in the same way to the ability to maintain attention [3].

Neurological disorders such as Pervasive Developmental Disorder (PDD), Asperger's Syndrome (AS) and Attention Deficit Hyperactivity Disorder (ADHD), among others, are a problem that has been present for years, which are disorders that affect life in all aspects of people, their personal life, their social development, their learning ability and motor control due to the characteristics of each disease that by their similarities tend to be confused due to their similarity in symptoms [4].

The correct detection can mean a great help in the life of the person who suffers from a neurological disorder, because they can take the appropriate psychiatric care throughout their life, allowing the medical treatment to allow them to lead a life without so much affectation by the disorder, since equally some additional manifestations to the cognitive ability are physical manifestations, seizures, vomiting, mental retardation, hyperactivity, etc. [5].

Saccadic movements are the ability of rapid displacement of the eyes changing their fixation from one point to another in a visual field. This ability is one of the most repetitive affectations in all disorders due to the poor control over the motor-ocular movements, when it comes to maintaining active attention [4, 5, 6].

That is why it is proposed in this work to develop an artificial intelligence model that allows to provide a more accurate means at the time of detecting the states of attention through eye movements as a prelude to the view of a specialist serving as a detection backup tool. As explained if people who suffers a neurological disorder, are the main beneficiaries of this research since it would provide a support tool for the correct detection.

2 Background

Identifying the previously elaborated works in all the areas of interest that the project covers is of utmost importance because it will help to get an idea of what advances have been made over time; in addition, it allows the identify areas of opportunity to improve and expand in the area of artificial intelligence on the topic this investigation aims; in this case, it will be necessary to identify the works previously done regarding the attention levels, how it has been related to artificial intelligence.

2.1 Cognitive Attention

Cognitive ability has gained relevance as an object of study, being approached from different angles, approaches, and through different research methods, it is an area of multidisciplinary interest encompassing psychology, neuroscience, education, and recently Artificial Intelligence (AI), to understand how humans process, store and use information [7].

Now focusing on AI, it is evolving significantly from early research to current applications using more advanced models, the union between AI and cognition has developed multiple approaches, from simulation, classification, and detection of cognitive processes to the development of models and systems that seek to replicate human intelligence [8, 9]. One of the most important cognitive processes is attention, which regulates the ability to process and direct mental resources toward relevant stimuli. Despite being an old subject of study, its measurement is still subjective since there are no standards to determine how much attention a person has, which is why there are different ways to classify it, from states of attention to levels of attention [10, 11].

2.1.1 Attentional States

In various aspects of life, states of attention are critical to optimizing cognitive performance, information retention, and decision-making. To improve safety, communication, and interpersonal relationships during critical situations, it is critical to take the necessary steps to maintain attention.

Attention is a topic of interest that has been investigated over time, although there is no standardization for its classification since it tends to be categorized into attentive, non-attentive [12, 13]. This classification is somewhat limited due to the existence of some external factors that interfere in its determination [3].

The classification proposed in the work of Abdelrahman [3] and in the work of Sohlberg and Mateer [14] both propose a broader grouping with five types of classification taking into account more factors, such as the possibility of the existence of some kind of deficit that would limit the person's ability to maintain attention. This presents a better way to describe each of the person's behaviors, without limiting it to his or her ability to be attentive or not to the task at hand.

Most research uses bioelectrical signals to classify attentional states, such as that developed by Kaushik et al [10]. Who focused their study on a system capable of distinguishing between attentional and distracted states, but using EEG-type signals. Another approach is the research developed by Yoo et al. [15] in which, using fMRI, they attempted to develop a set of whole-brain models to predict different attentional states as well as memory capacity.

Using eye trajectory data, Bello et al [16]. Addresses a problem in estimating and classifying attentional states. The research proposes a methodology in which one-dimensional data are transformed into two-dimensional data to have a clear perspective of how the person's attention is, applying deep learning (DL) techniques, high performance in classifying attentional states was demonstrated.

2.1.2 Attention Levels

AI has been a supporting tool in which several investigations have been developed to measure, classify and detect levels of attention. A person's level of attention refers to the degree of cognitive activation that a person experiences when processing stimuli through the senses, as opposed to attentional states, the level of attention indicates how much of it is available at a given moment.

Several studies have been developed during these last years, one of them is the one led by Samei et al. [17], in which they tried to predict using a Convolutional Neural Network (CNN) visual attention and distraction during visual search activities, this research provides an understanding of the attentional cognitive process with the integration of artificial systems.

When attempting to simulate a person's level of visual attention, Hazan et al [18]. Introduced a model based on Recurrent Neural Networks (RNN) combined with reinforcement training, which classifies visual stimuli by learning to attend to specific regions of an image, helping to understand the cognitive process that is followed for attentional mechanisms.

Through a specific test to determine attention (ADEM), Garcia et al [19]. Developed a methodology based on CNN in which a person's attention levels were classified as a tool for the detection of cognitive disorders, demonstrating the effectiveness of AI in the classification using specific tests to estimate attention.

These studies show that attention levels have become relevant thanks to the combination with AI, with the intention of contributing to the understanding and optimization of attention in humans. As AI advances, it is expected that this research will take more advanced degrees of complexity, providing more tools in the field of cognitive processes.

2.2 Saccadic Movements

The position of the eyes becomes relevant because, as Hoffman [20] demonstrated in his experiment, a person can't move the eyes to one focus and be attentive to a different one,

therefore the investigation infers that saccadic movements are of utmost importance for visual attention since the intention to pay attention voluntarily generates a saccadic movement.

2.2.1 Eye-Tracking

Given that the eyes are usually associated with attention, due to the perception that they have over time, to make an analysis and make the detection, you can determine the states of attention using an eye tracker, this works by using the position of the eye following its trajectory building in most cases vectors, which serve to analyze their behavior [21].

When people suffer from any type of deficit, some indication of the condition can be detected because one of its symptoms is not being able to have effective control of saccadic movements, in this way specialists can determine what type of attention the person has [22, 23, 24].

There are different techniques of use of eye tracking, from the use of glasses which have an integrated camera in which they are doing the tracking [22], another technique of use is in which a specialized camera is used to follow the trajectory of the eyes, without being an invasive method since it is only necessary that the person does the test that is required [3].

The different eye tracking techniques generate vectors in space (x,y), which allow us to know the trajectories and fixation points, when attention is maintained on a fixation point it can be inferred that attention was focused on that particular point, according to research by Deans[25], fixation points last approximately between 200-300 milliseconds, consequently after that time it is already considered a fixation point.

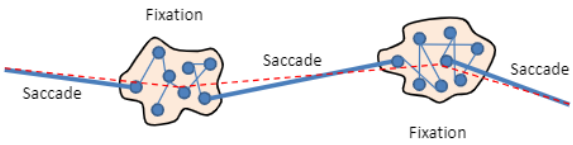


Figure 2.1: Eye tracking trajectories obtained [26].

Fixation points can be determined as circumferences that grow, depending on the time that attention is maintained [26], which is why a relationship is maintained between fixation points and saccadic movements with attentional states.

2.3 Machine Learning

In the area of artificial intelligence, machine learning (ML) has been widely used to analyze in different ways the cognitive capabilities that serve to demonstrate how it can contribute to improving cognitive understanding, offering new perspectives in this area.

One of the most recent studies is that of Vázquez et al. [27], who used Large Language Models (LLM), to demonstrate that this type of model can develop executive functions similar to those of humans, with the Towers of Hanoi test they evaluated different aspects of how the model planned and memorized the sequence, demonstrating that LLM are capable of developing executive functions.

Using eye tracker data, Bedolla et al [11]. Developed an investigation in which it was proposed to classify the attention levels of a person when performing psychometric tests using a random forest model optimized with genetic algorithms, providing a tool that serves to estimate when a person may suffer from a cognitive disorder.

Merchan et al [28]. Proposed an architecture intended to model human consciousness in an artificial system, with the hypothesis that with sufficient information resources, an intelligent system is capable of making coherent and accurate decisions, suggesting that by adapting Gaussian processes in conjunction with deep learning (DL) techniques, it is possible to build agents that increasingly simulate more complex processes.

2.4 Related Works

Although this research is focused on determining the levels of attention using eye trajectory data, more works are in charge of classifying these levels but using data of a different nature, although these techniques are mostly bioelectrical signals, some of the most relevant ones with these types of data are presented.

Electroencephalograms (EEG) can also be used to determine a person's state of attention, but it is usually a highly invasive method, which consists of studying the brain waves depending on the activity they are performing to determine the person's potential state of attention. In the research carried out by Chen[29] and Toa [30] it can be seen that they use artificial intelligence models to process the signals and give an estimate of the type of attention.

Belle[31] in his work, proposed to determine the states of attention using electrocardiograms, which proposed to use a monitor to determine the lack of attention of the person when executing a task, to be able to relate the signals captured by the Electrocardiograph and the person's attention. Using the Stockwell transform, they decompose the signals to be analyzed using a machine-learning model.

2.4.1 State of art

The state-of-the-art delves into previous research that focuses on the classification of attention levels, Table 2.1 mentions previous work related to the classification of attention levels, using eye-trajectory data as well as bioelectrical signals.

Main author	Article name	Database characteristics	Classification	Model	Metrics (Accuracy)
Eye Tracking					
García et al. 2024 [19]	Clasificación de Niveles de Atención con Datos de Desempeño y de seguimiento Ocular Durante la Prueba ADEM	55 unspecified subjects	Under, Fair, Good, High	Fully connected artificial neural network	95.82%
Bedolla et al. 2022 [11]	Classification of attention levels using a Random Forest algorithm optimized with Particle Swarm Optimization	22 females, 33 males	Low, Medium, High	Random Forest	92.5%
Barz et al. 2021 [32]	Automatic visual attention detection for mobile eye tracking using pre-trained computer vision models and human gaze	25 people	Detection, Not detection	ResNet	84%
O. T. Chen et al. 2017 [21]	Attention Estimation System via Smart Glasses	10 people	Attentive, Non-attentive-	G(GA)-SVM	93.10%
Different techniques					
Toa et. al. 2021 [30]	Electroencephalogram-Based Attention Level Classification Using Convolution Attention Memory Neural Network	18 males, 12 females	Attentive, Non-attentive	CAMNN	92%
Chen et. al. 2017 [29]	An automated optimal engagement and attention detection system using electrocardiogram	10 People	Attentive, Non-attentive	SVM GA-LIBSVM	89.52%

Table 2.1: State of the art

3 Theoretical Foundation

3.1 Cognitive Attention

Is a process that gives our brain the ability to select, concentrate and process relevant stimuli from the environment while ignoring irrelevant stimuli, it is one of the most important cognitive processes because it regulates perception, memory, decision making and has an impact on human behavior.

Attention is not a unitary process, but a set of processes working together to perform the activity, in addition, it is a limited resource since it is adjusted according to the cognitive demand and ability to perform a task, the main characteristics are, selectivity to focus on a specific stimulus disregarding external stimuli; voluntary control to direct the focus of attention to stimuli; persistent, since it can be sustained over time, although it can be affected by fatigue or distractions; alternation, since it allows changing the focus of attention between different tasks or stimuli according to the will.

It is an essential part of people's daily lives, since it allows them to adapt to the environment and be reactive in an efficient way to stimuli; in short, cognitive attention is a complex process that regulates the way in which individuals process information.

A derivative of cognitive attention can be the states of attention, and the levels of attention, since as mentioned it is the ability to process stimuli provided by our senses, in this case when we want to detect the states of attention visual stimuli are prioritized, since a person with an identified cognitive impairment, can affect the sensory capacity of the sense of sight, impacting directly because the same deterioration of the ability to process stimuli would influence their state of attention [33].

Types of Attention	**Criterion**
Selective, Divided, Sustained	Participating Mechanisms [3, 34]
Visual, Auditory	According to Sensory Stimulus [35]
Global, Selective	Amplitude and Intensity of Attention [36]
Voluntary, Involuntary	Degree of Voluntary Control [37]

Table 3.1: Classification of Attention Types Based on Different Criteria

According to psychology and neuroscience, there are different types of attention, table 3.1 represents these types of attention and the criteria by which they are classified, this is useful to understand how attention works in the brain, each type fulfills a specific function in the processing of information.

3.1.1 Attentional States

Attention is a multifaceted cognitive process encompassing various dimensions. Among these, sustained, alternating, selective, and divided attention represent distinct ways individuals engage with and process information [38, 39].

a. **Sustained Attention**: Often referred to as vigilance or focused attention, sustained attention pertains to the ability to maintain an unwavering focus on a single task or stimulus for an extended duration. It is critical in endeavors requiring prolonged concentration, such as reading a book, viewing a movie, or monitoring a control panel [38, 3].

b. **Alternating Attention**: Alternating attention embodies the skill of adeptly shifting focus between multiple tasks or stimuli. It entails the art of seamlessly transitioning between diverse activities and data sources. This skill is especially invaluable in tasks demanding flexibility and adaptability, such as multitasking or swiftly changing work scenarios [3].

c. **Selective Attention**: Selective attention channels concentration onto a specific stimulus while concurrently disregarding or sieving out other distractors or irrelevant data. It proves pivotal in activities like listening to a speaker amid background noise, reading in a bustling setting, or watching a film in a lively environment [38].

d. **Divided Attention**: Divided attention, commonly recognized as multitasking, revolves around the ability to allocate one's focus to numerous tasks or stimuli concurrently. It entails the capability to engage in two or more activities simultaneously without experiencing a marked decline in performance. This proficiency finds applicability in scenarios where managing various responsibilities is requisite, such as driving while conversing or typing while absorbing a lecture [38].

These distinct facets of attention feature prominently in myriad aspects of daily life, with the potential for enhancement through practice and training. The proficiency levels in each type of attention can fluctuate from person to person and can be influenced by factors like age, cognitive aptitude, and intentional training efforts [40].

3.1.2 Attention Levels

It can be defined as the visual selection and inspection of regions of interest, it is a mechanism that allows the discrimination of relevant information and the inhibition of other stimuli [41], in other words, it refers to the hierarchy of visual processing in the brain.

The levels of attention can be classified into three general categories according to the research of Ballesteros [42] and Posner et al.[43]:

- **Low level:** Also known as automatic or pre-attentional attention, it is the unconscious level of attentional processing, one of its main characteristics is that it occurs in milliseconds the change of focus and does not depend on the will of the individual, it is activated quickly and automatically when a stimulus different from the one being focused is presented.

- **Medium level:** It is the level of focused or selective attention, the brain can process the information focusing on the relevant ones ignoring most of the time the irrelevant ones, it requires more effort and cognitive processing, although the attention is consciously controlled, it usually has certain deviations of attention.
- **High level:** The most complex and controlled level of attention, it can also be called Executive Attention, in which more cognitive processes such as memory, planning and decision making are involved, it is required for tasks that involve thinking, learning and control of the focus of attention voluntarily.

Characteristic	**States of Attention**	**Levels of Visual Attention**
Definition	Different ways in which attention shows.	Hierarchical processes of visual analysis in the brain.
Focus	Distribution and control of attention over time.	How visual information is processed from basic to complex levels.
Flexibility	Changes depending on the task and context.	More structured, based on the organization of the visual system.

Table 3.2: Summary of the Differences Between States of Attention and Levels of Visual Attention

Table 3.2 represents the differences between states of attention and levels of attention, explaining the main characteristics of each, thus it can be concluded that the states of attention can be present in the levels of attention, since it is the way in which the cognitive process is being performed while the other process is the amount of attention that is being paid to a task.

3.2 Saccadic Movements

Commonly known as saccades are the rapid and automatic eye movements that are a fundamental part of our visual perception or, in other words, of our visual attention. These eye movements are essential when interpreting the stimuli in our environment. Saccades facilitate the rapid transition from one focal point to another, speeding up the process of visual exploration, usually more evident in activities such as reading, since these saccadic movements transport attention quickly from one word to the next, allowing us to process textual information [20, 44].

A characteristic of saccades is the use of the entire visual field, taking advantage of peripheral vision, through these rapid and practically imperceptible movements a

navigation through the entire visual field is produced, allowing to obtain information from different points of view. When the eyes stop on specific objects or points of interest, saccadic movements acquire another name, fixations, which offer an opportunity to deepen the visual details of a specific point of our environment [45].

3.3 Artificial Intelligence

Is the emulation of human-like intelligence and cognitive processes in machines [46], AI encompasses a wide range of techniques that enable intelligent systems to perform tasks typically associated with human intelligence, including learning, reasoning, decision making and natural language processing. AI uses mathematical models and algorithms to process data and make decisions based on different techniques:

- **Machine Learning (ML):** These are algorithms that learn patterns of data.
- **Artificial Neural Networks:** Models inspired by the human brain.
- **Natural Language Processing (NLP):** A type of technique where machines are trained to understand human language.
- **Computer Vision:** It is the analysis of images or videos for different specific tasks.

AI has been evolving over the years, becoming applicable in more and more areas, including medical, robotics, economics, education, etc. Focusing AI on attention, relating it to cognitive capacity, it has a highly wide applicability due to the detection of small patterns when processing large amounts of information, providing greater accuracy and efficiency in the processing of information compared to human capacity [47, 48].

3.3.1 Machine Learning

It is an essential coding technique that is focused on learning from data through the application of algorithms and statistical models. Unlike traditional programming, its objective is to develop systems that improve performance by means of learning from data; it consists of training models to identify patterns and, based on that, make independent decisions. The general idea of this technique is to be able to generate models capable of generalization, that are efficient in the face of data with which they were not trained and can continue to improve[48].

In practical applications, ML incorporates a multitude of techniques and models ranging from neural networks, decision trees, support vector machines, logistic regression, etc. To more complex models such as Ridge Classifier, Extra Trees Classifier, Gradient Boosting Classifier, which are more complex models based on the simple models [49].

These versatile models are used in different tasks such as image recognition, autonomous vehicle recommendation systems, medical diagnosis, and others. Through the use of data, ML provides the different algorithms with the ability to capture specific patterns, allowing them to evolve as more information is generated and improve the functionality of the model [49].

Model	Characteristics	Advantages	Disadvantages
Linear Regression [50]	Predicts the behavior of a variable based on another by fitting a straight-line relationship.	Easy to interpret and train. Useful for linearly related data. Practical for binary classes.	Does not adapt to non-linear data. Sensitive to outliers. Overgeneralizes.
Logistic Regression [51]	Statistical model used for classification.	Fast, efficient, and simple for binary classification tasks. Easy to interpret results.	Assumes independence among predictors.
Decision Trees [52]	Hierarchical rule-based model that splits data into branches, mimicking a tree structure.	Easy to interpret. Handles categorical and numerical variables.	Prone to overfitting if not well-configured.
Support Vector Machines (SVM) [53]	Finds the optimal hyperplane to separate classes in classification tasks.	Works well with high-dimensional data. Suitable for small datasets.	Difficult to interpret.
Gradient Boosting (XGBoost, LightGBM) [54]	Models that iteratively improve decision tree ensembles.	High performance in classification and regression. Less prone to overfitting.	Computationally expensive.
Extra Trees Classifier [55]	Constructs multiple decision trees and averages the results while randomly selecting pruning points.	Reduces variance by adding extra randomness to the model. Computationally efficient.	Difficult to interpret.
Ridge Classifier [56]	Modified linear regression that incorporates optimization techniques and can be used for classification.	Simple and efficient. Suitable for high-dimensional datasets.	Requires variables to be on the same scale to avoid bias toward certain features.

Table 3.3: Comparison of Machine Learning Models

Table 3.3 presents the most common ML models, addressing the characteristics of each one, as well as the advantages and disadvantages of working with each model, giving an idea of the type of scenarios in which they can be selected, in addition to the optimized models of the basic models, which, despite having a high computational cost, aim to improve the results.

Logistic Regression It is a statistical model designed for classification, and although the name indicates regression is not used for that, it is based on the probability of belonging to a class of a category [51]. When more than two classes are classified, special techniques called One-vs-Rest (OvR) or Softmax (Multinomial Logistic Regression) are used [57], the latter being the one used in the project.

To estimate the probability of each observation belonging to a specific category, the model estimates the probability of each class and assigns the label to the class with the

highest probability of being. Mathematically it can be explained as follows, if there is a dataset with X characteristics and a target Y with three classes:

$$Y \in \{A, B, C\} \tag{1}$$

When the OvR technique is used, the model assigns the probabilities using the sigmoid function, while if the classification technique is Softmax [58], they are assigned as follows:

$$P(Y = C_k|X) = \frac{e^{\beta_k X}}{\sum_{j=1}^{3} e^{\beta_j X}} \tag{2}$$

Where:

- $P(Y = C_k|X)$ is the probability that the observation belongs to class C_k.
- β_k are the logistic regression coefficients for class k.
- e is the base of the natural logarithm.

Method	**Functionality**	**Advantage**	**Disadvantage**
One-vs-Rest (OvR)	Converts the classification problem into three separate binary classification problems and assigns the label with the highest probability.	Easy to implement, adapts well for problems with well-defined classes that do not resemble each other.	May cause inconsistencies in probability estimation among models, does not capture relationships between classes.
Multinomial Logistic Regression (Softmax)	Generalizes logistic regression using the softmax function, assigning a probability to each class and selecting the one with the highest value.	Produces well-calibrated probabilities, adapts well when the data classes are similar.	Requires higher mathematical and computational cost, susceptible to bias.

Table 3.4: Comparison of Multiclass Classification Methods

Table shows 3.4 the differences between the two multiclass classification methods, demonstrating what each consists of, their advantages as well as their disadvantages, these two methods allow to adapt a model designed for binary classification to more classes.

Hyperparameters Are variables that control the training of ML models, these are external configurations to the model that have to be determined before training the model, each model has its own hyperparameters because the architectures of each one are different [46].

Multinomial Logistic Regression with the Softmax method has adjustable hyperparameters [59] that affect the training and generalization capacity of the model, the most important of which are presented below:

a. **Optimization parameters**

- **Solver:** An algorithm is assigned to optimize the optimal coefficients of the model, among the most common solvers are lbfgs, which is recommended for small and medium datasets; Saga, which is an adaptation of the Stochastic Avarage Gradient Descent method, with the difference that it supports both types of regularization (L1 and L2); newton-cg, which uses Newton's method for optimization, and is ideal for convex problems.
- **Regularization Strength (C):** It is in charge of controlling the penalty applied to the coefficients of the model with the intention of avoiding overfitting, a high value of C reduces the regularization, while a low value increases it, the value of C is the inverse of the regularization.

b. **Regularization parameters**

- **Penalty:** Specifies the type of regularization used during training, having as methods ridge regression (l2), which is the one used by most solvers; lasso regression (l1) forces coefficients to be zero; elasticnet, which is the combination of both l1 and l2.

c. **Convergence and accuracy parameters**

- **Maximum Iterations (max_iter):** Defines the maximum number of iterations before the model converges.
- **Tolerance (tol):** Specifies the stopping criterion basing the previous result with the current one during several iterations with a defined threshold, having a small value may increase the training time.

d. **Additional parameters**

- **Multi-class (multi_class):** Specifies the classification method for problems with more than two classes, having multinomial using the softmax function and ovr generating a binary classifier for each class.

The above list presents the most important hyperparameters when working with multiple classes, but not all variables are configurable. Table 3.5 presents all possible hyperparameters when working with the Logistic Regression model according with

Scikit-Learn documentation [59], presenting their description, their possible values and the default value in the model.

Hyperparameter	**Description**	**Possible Values**	**Default Value**
solver	Optimization algorithm to find the optimal coefficients.	'"lbfgs"', '"sag"', '"saga"', '"newton-cg"'	'"lbfgs"'
C	Controls model regularization (inverse of λ). A higher value reduces penalty.	Positive values ($C = [0.01, 0.1, 1, 10, 100]$)	'1.0'
penalty	Type of regularization applied to the model.	'"l2"', '"l1"' (only with '"saga"'), '"elasticnet"', '"none"'	'"l2"'
max_iter	Maximum number of iterations for the model to converge.	Positive integers ('100', '500', '1000')	'100'
tol	Tolerance for convergence criterion.	Small values ('1e-4', '1e-3', '1e-5')	'1e-4'
multi_class	Defines how multiclass classification is handled.	'"multinomial"', '"ovr"' (One-vs-Rest)	'"auto"'
fit_intercept	Determines whether to include an intercept term in the equation.	'True', 'False'	'True'
class_weight	Adjusts class weights in case of imbalance.	'None', '"balanced"'	'None'
intercept_scaling	Scaling factor for the intercept term (only for 'liblinear').	Positive values	'1.0'
verbose	Level of detail in training output.	'0' (silent), '1' (minimal details), '2' (extended details)	'0'
n_jobs	Number of processors to use for training.	'None', '-1' (uses all available cores)	'None'

Table 3.5: Hyperparameters of Multinomial Logistic Regression

3.3.2 Metrics

Evaluation metrics are tools to measure the performance of the models, depending on the type of task being performed, for classification problems the model is evaluated with the ability to assign the labels correctly to each class, it means, that the predictions are according to the actual labels, these metrics are based on the confusion matrix, which is a table that shows the times that a model correctly or incorrectly classified the instances.

Confusion matrix is a table that shows the classification performance by comparing the predictions with real labels, it is mainly used in binary and multiclass classifications, through the confusion matrix it is possible to estimate more metrics including Precision, Recall, F1-score, Accuracy, and more advanced metrics such as Kappa and MCC [60].

It also provides visual information of the model predictions, giving results such as True Positives (TP), which are the correct predictions of a specific class; False Positives (FP), which incorrectly predict a class when it does not correspond; True Negatives (TN), which in this case are the correct predictions of classes other than the one of interest; False Negatives (FN), which predict another class when it should have been the correct one. Figure 3.1 gives a visual representation of what a Confusion Matrix looks like for three classes.

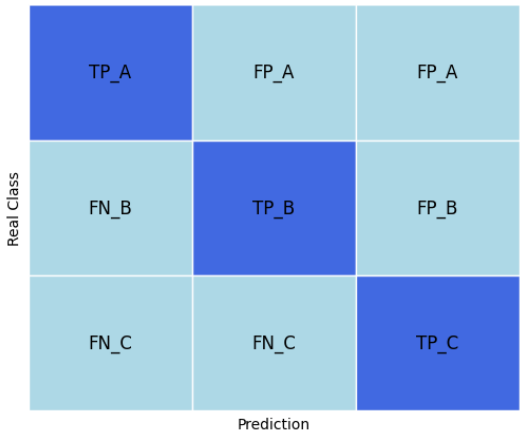


Figure 3.1: Confusion Matrix. Adapted from [61]

Accuracy It is a metric that measures the percentage of correct predictions, it is useful when the classes are balanced [60], to calculate the accuracy it is done as follows:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{3}$$

Precision It measures how many of the positive predictions were really well predicted, this metric takes a lot of importance when medical diagnosis is being performed since it considers giving much weight to false positives [60], it is estimated:

$$Precision = \frac{TP}{TP + FP} \tag{4}$$

Recall Indicates how many positive cases were correctly detected, being a metric that considers false negatives is ideal when working with disease detection, frauds, etc. [60]. It is calculated as follows:

$$Recall = \frac{TP}{TP + FN} \tag{5}$$

F1-Score It is the harmonic average between precision and recall [60], since it is calculated considering these two metrics:

$$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{6}$$

The previous metrics are useful when the classes are balanced or there is not much difference between them, but when you have datasets with unbalanced classes you have more advanced metrics that serve to evaluate the performance of the models.

Kappa Cohen (κ) It measures the degree of agreement between the model and the prediction, it is calculated as shown in the equation, depending on the value obtained is the level of agreement, if it is close or less than zero, it would be considered a poor agreement, and the closer it is to one of as value an almost perfect agreement [62].

$$\kappa = \frac{P_o - P_e}{1 - P_e} \tag{7}$$

where:

$$P_o = \frac{TP + TN}{TP + TN + FP + FN} \tag{8}$$

$$P_e = \frac{(TP + FP)(TP + FN) + (TN + FN)(TN + FP)}{(TP + TN + FP + FN)^2} \tag{9}$$

Matthews Correlation Coefficient (MCC) The MCC measures the quality of a classification considering all the values of the confusion matrix, this metric is especially useful when you have unbalanced classes, since it evaluates the classification considering both true and false values [63], it can be calculated as follows:

$$MCC = \frac{(TP \times TN) - (FP \times FN)}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}} \tag{10}$$

3.3.3 Neural Networks

Also called artificial neural networks (ANN), they are a class of models belonging to machine learning with an architecture inspired by the structural and functional dynamics of the human brain. These networks are the fundamental basis of deep learning, and one of their main characteristics is the ability to learn autonomously and the capacity to generalize data [64, 65]. Within a neural network, there are structures of node names or artificial neurons that are usually interconnected and organized by means of capable, usually divided into three main categories:

a. **Input Layer**: This initial layer acts as the receiver of the data from the data set, each input value is represented by a neuron or particular attributes of the data.

b. **Hidden Layers**: The intermediate layer is the one that exists between the input layer and the output layer, these perform the union between the layers, neural networks can be made up of multiple intermediate layers or also called hidden, and this stratification is what is designated "deep layers", which contributes to the designation in deep learning.

c. **Output Layer**: The output layer furnishes the final results or predictions after processing the data in the hidden layers. The number of neurons in this layer varies according to the specific task, such as binary classification, multi-class classification, or regression.

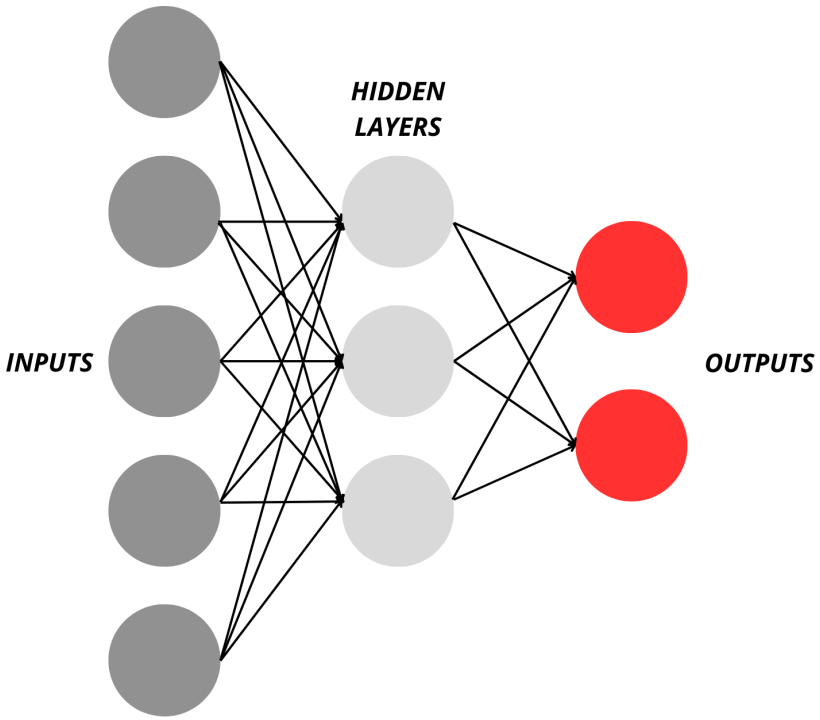


Figure 3.2: Architecture of a neural network. Adapted from [64].

Neural networks are trained through a mechanism called supervised learning, where the weights and biases of the connections between neurons are adapted to minimize the mismatch between the network prediction and the objective values of the training data set. This training procedure usually employs the backpropagation technique. ANNs are flexible and can be applied in tasks such as image recognition, class classification, value prediction. Their prowess lies in their ability to understand patterns and complex representations from data, which makes them an indispensable component of artificial intelligence and machine learning [66].

Convolutional Neural Networks One area of ANNs is Convolutional Neural Networks (CNNs), which are a type of network specially designed to handle visual data, especially images. They are known for their ability to autonomously and adaptively perceive spatial hierarchies of features within images used as input data. The architecture of these layers is made up of different components: convolutional layers, grouping layers and fully connected layers [67, 68].

Convolutional layers are a fundamental part of CNNs and are responsible for detecting patterns and local features in the input images, they employ convolution operations, which is a process in which small filters and kernels traverse the image to discern features such as edges, textures and shapes. The next stage is the clustering layers, which serve to reduce the size of the data and improve computational efficiency, the most common pooling techniques are max-pooling and average-pooling. Once the image is minimized after having passed through all the layers that the CNN has, the fully connected layers are employed which is normally the culmination of the network, at this stage the features previously detected and extracted by the preceding layers are gathered and amalgamated, preparing the data for the tasks of classification, regression, detection or segmentation [69].

3.4 Data Augmentation

It is a technique within ML and DL used to generate synthetic data from the original data, its objective is to improve the generalization of the models since it increases the quantity and diversity of the training data, thus seeking to reduce problems such as overfitting or underfitting [70].

Data augmentation is based on transforming the original data to create new instances without altering the main characteristics of these, depending on the type of data to be worked on are the techniques that can be applied.

There are different techniques for each type of data, from images, text, audio, to numerical data in general, each one has different techniques with its degree of complexity to ensure a successful data augmentation. Image transformations can be divided into two subgroups, Geometric Transformations and Color Transformations, which in turn each have different techniques to transform the data, including rotation, scaling, translation, flip, in the case of Geometric, and brightness adjustment, conversion to color scale, Gaussian noise for when it comes to color transformations [71].

When augmenting data, it is not enough just to apply arbitrary transformations, but it is crucial to validate the augmented images to ensure that they are useful and that noise or irrelevant images are not being introduced. Visualization techniques and metrics that give us numerical values must be used to measure the quality of the augmented images before training a model.

3.4.1 Diversity

It refers to the measure of variation between the original data and the synthetic data, it is a fundamental metric in the validation of the augmented data, since it allows to evaluate the different points within the feature space.

To estimate and then evaluate the diversity in the generated data, the Euclidean distance between the augmented images and the original images must be calculated; this demonstrates whether the synthetic data set is within an acceptable range of variation. The Euclidean distance is calculated as follows:

$$\text{Diversity} = \frac{1}{\binom{N}{2}} \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} \|x_i - x_j\| \tag{11}$$

Where $\binom{N}{2} = \frac{N(N-1)}{2}$ is the total number of possible combinations of two distinct elements of the data set, thus ensuring that each pair (i, j) is counted only once [58].

This version of the metric is frequently used in analysis to assess the dispersion of points within a group. Using unordered combinations avoids counting duplicates, which provides an accurate and consistent representation of the internal variability of the data [72]. In the context of machine learning, measuring the diversity of a data set is critical to assess whether the points generated using data augmentation techniques are sufficiently varied, which can improve the generalization of the model [50].

3.4.2 Affinity

The affinity is a metric used in the validation of data augmentation, it is used to evaluate how similar the augmented data are to the original data, the intention of this is to ensure that after having generated synthetic data the classes have not suffered an excessive alteration in the distribution of the data, but on the contrary, that they maintain consistency in the important characteristics to ensure data quality to the model. It can be calculated by means of different auxiliary metrics that compare the original data with the augmented data, these metrics are:

- **Calculate the Average of Characteristics per Class:** For each class c, it calculates the average of the features of all images belonging to that class:

$$\bar{x}_c = \frac{1}{N_c} \sum_{i=1}^{N_c} x_i \tag{12}$$

where x_i is the feature vector of the image i in the class c, and N_c is the number of images in the class c.

- **Calculate the Average Distance per Class:** The affinity is calculated as the average of the Euclidean distances between all pairs of images in the same class:

$$\text{Affinity}_c = \frac{1}{N_c(N_c - 1)} \sum_{i=1}^{N_c} \sum_{j \neq i} \|x_i - x_j\| \tag{13}$$

where $\| \cdot \|$ represents the Euclidean norm.

- **Average Affinity:** The overall affinity is obtained by averaging the affinity of all classes:

$$\text{Affinity}_{\text{avg}} = \frac{1}{C} \sum_{c=1}^{C} \text{Affinity}_c \tag{14}$$

Where C is the total number of classes.

When calculating the affinity, the value obtained must be evaluated, where a value close to zero indicates the similarity of the images within the same class, this value close to zero denotes that the images are consistent and homogeneous, which guarantees a good training. The affinity is low when the value obtained is close to 1, indicating that the images within that class do not conserve the essential characteristics of the class to which it belongs [73].

3.4.3 t-SNE

It is a nonlinear dimensionality reduction method that is mainly used to visualize data in lower dimensional spaces such as 2D and 3D. The goal is to preserve the structure of the data by reducing the dimensionality [74], the algorithm works as follows:

a. Calculates the probabilities of high dimensionality data.

b. Calculates the probabilities in low dimensionality

c. Minimizes differences between distributions.

t-SNE converts the high-dimensional Euclidean distances into conditional probabilities that represent the similarity between two points, what the technique tries to do is to minimize the divergence between the two probabilities calculated and already

mentioned, one measures the similarities in the high-dimensional space and the other measures them in the low-dimensional space. The objective function is defined as:

$$C = \sum_{i} \sum_{j \neq i} P_{ij} \log \frac{P_{ij}}{Q_{ij}} \tag{15}$$

Where:

- P_{ij} represents the similarity between high-dimensional points x_i and x_j.
- Q_{ij} represents the similarity between low-dimensional points y_i and y_j.

The similarity P_{ij} is calculated using a Gaussian distribution centered at point x_i:

$$P_{j|i} = \frac{\exp(-\|x_i - x_j\|^2 / 2\sigma_i^2)}{\sum_{k \neq i} \exp(-\|x_i - x_k\|^2 / 2\sigma_i^2)} \tag{16}$$

where σ_i represents the variance of the Gaussian centered on x_i.

The low-dimensional similarity Q_{ij} is defined using a Student t-distribution with one degree of freedom:

$$Q_{ij} = \frac{(1 + \|y_i - y_j\|^2)^{-1}}{\sum_{k \neq l} (1 + \|y_k - y_l\|^2)^{-1}} \tag{17}$$

It is a useful technique to be able to identify patterns, clusters, in the approach of data augmentation validation, it serves to have a visual perspective of the distribution of the synthetic data over the original ones, allowing to observe the quality of these new data.

3.5 Optimization

Optimization is an essential process when developing models, it consists of finding the best set of values in the hyperparameters for a specific model, in order to maximize its performance and improve its metrics in the task it is performing. The function of an optimization is to reduce overfitting and underfitting, to adapt the model to specific characteristics of the data set, to guarantee computational efficiency, since it has the values in which the models take full advantage of all the resources to deliver the best results.

Table 3.6 presents the different optimization techniques, where each one has its advantages and disadvantages, in addition to other selection criteria, such as model complexity, computational capacity, if you want to find the optimal values, however, a correct optimization generates more robust models and in turn produces more reliable results.

Method	Description	Advantages	Disadvantages
Grid Search [75]	Tests all combinations in a defined grid.	Exhaustive, easy to implement, and easy to understand.	High computational cost as the number of variables increases.
Random Search [76]	Randomly selects combinations within a defined range.	Efficient in large spaces with many variables.	Does not guarantee finding the best combination.
Bayesian Optimization [77]	Selects combinations through probabilistic models that could be optimal.	Less computationally expensive.	Requires advanced probabilistic models.
Genetic Algorithm Search [75]	Seeks to find the best values through metaheuristic models or bio-inspired systems.	Explores solutions effectively.	High computational cost and time, requires tuning evolutionary parameters.

Table 3.6: Comparison of Hyperparameter Optimization Methods

3.5.1 Grid Search

It is an optimization method that, as mentioned in the table, explores all possible combinations of a set of defined variables, selecting the best configuration of hyperparameters according to the performance of the model [78]. To perform an optimization using Grid Search, the following must be done:

a. Generation of the grid: The matrix with all the possible combinations is generated.

b. Model evaluation: The model is trained and validated with each combination of grid values.

c. Model optimization: Once the entire grid has been run based on a defined metric (Accuracy, F1-Score, etc.) the best combination of hyperparameters is determined.

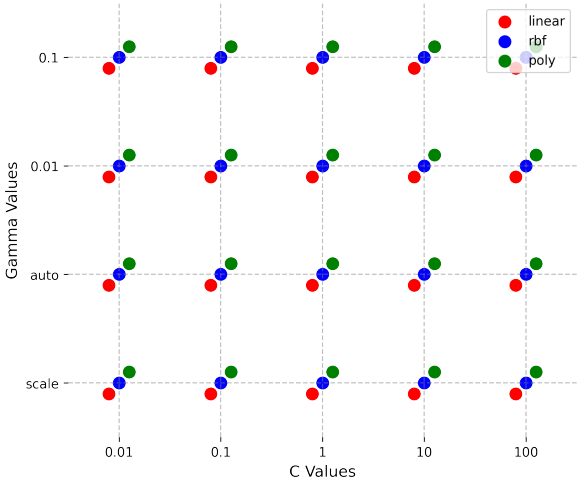


Figure 3.3: Grid Search representation

Figure 3.3 is a visual representation of what a Grid Search optimization looks like, where it can be seen that it is an exhaustive search, easy to implement and with reproducible results.

Where three specific hyperparameters are required to be optimized, the C value, gamma and kernel, where the values with the most possibilities (C and gamma) become the axes and at each intersection the three kernels are tested, guaranteeing to test all possible combinations of the set of values.

4 Hypothesis

An artificial intelligence model trained and optimized on eye-tracking data proposes an alternative method to improve attention level detection based on analyzing eye movement patterns, including fixations and saccadic trajectories.

5 Objectives

5.1 General Objective

Identify, validate, and select a suitable artificial intelligence model for detecting attention levels in a specific dataset, evaluating its accuracy using appropriate metrics.

5.2 Specific Objectives

- Collect and select different available and applicable databases for model training and validation.
- Identify the different artificial intelligence models applicable for detecting attention states.
- Clean, preprocess, and normalize data to ensure data quality and consistency.
- Train and validate the selected artificial intelligence models with the preprocessed data.
- Select and optimize the appropriate artificial intelligence model for attention level detection.

6 Methodology

Figure 6.1 is the visual representation of the workflow to accomplish the project.

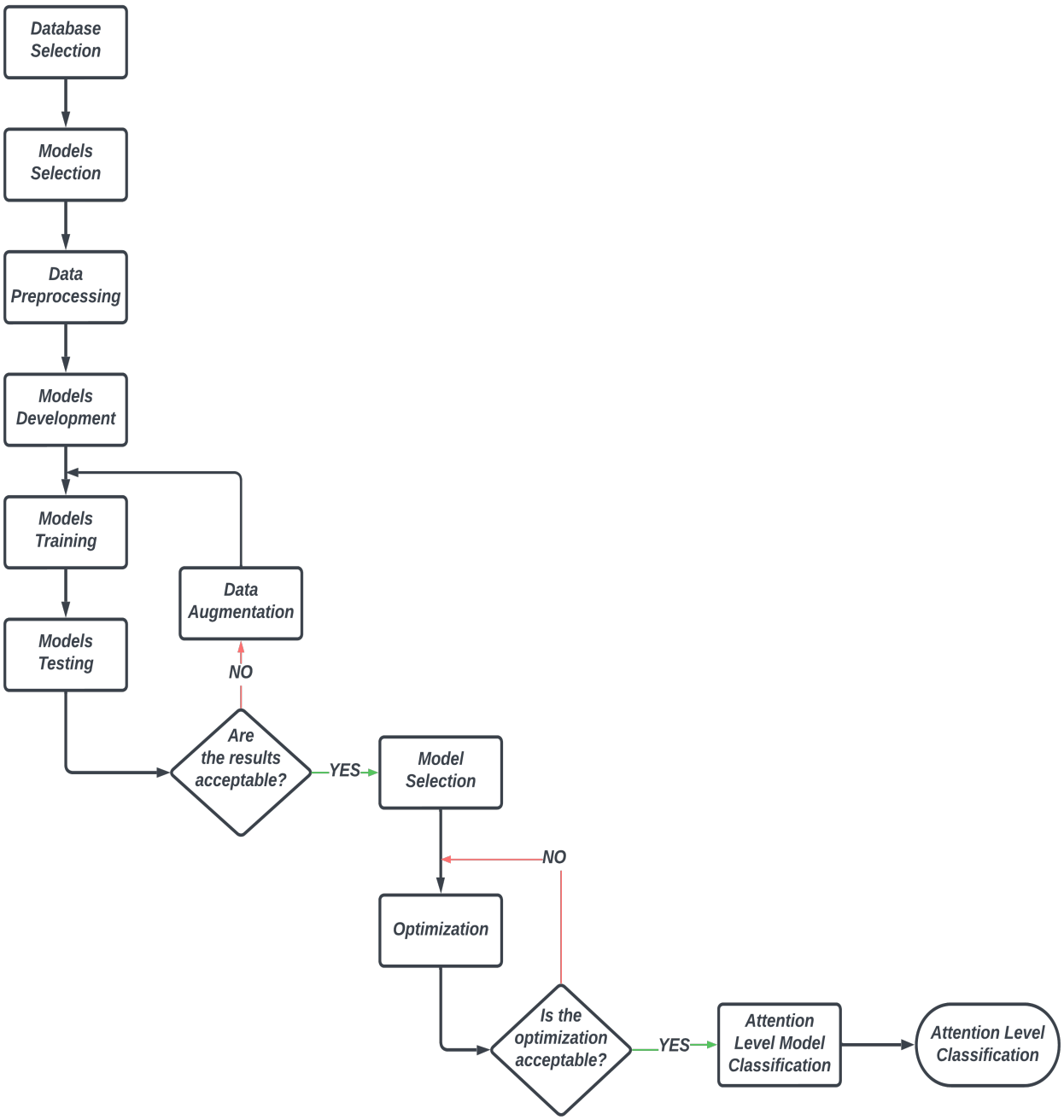


Figure 6.1: Methodology flowchart

6.1 Materials

A Legion computer[79] with an Intel(R) Core(TM) i7-8750H processor and 8GB of RAM was used. The project was developed using Python, and two different platforms were used for programming: Jupyter Notebooks[80] for the preprocessing stage and Google Colab[81] for the model development stage due to the accessibility of development environments in the cloud.

In addition, Google Colab used its premium subscription, working with an NVIDIA A100 with 80GB HBM2e GPU memory. Due to its high computational capacity and execution speed, it was used to train, validate, and optimize the models.

The libraries used in the project were Numpy, Matplotlib, Pandas, Pycaret, Random, Seaborn, Sclearn, and Tensorflow [82, 83, 84, 85, 86, 59, 87, 88], which served as support for the development of all the work, from data visualization and image generation to the training and optimization of the models. The libraries played a fundamental role in the development of this project due to their ease of implementation and the time saved by using them.

6.2 Database Selection

Selecting an adequate database is of the utmost importance since it is the data that will be used to train the model. The aim is to evaluate public databases published in congresses, journals, competitions, or websites to determine which best aligns with the project's purposes. Table 6.1 shows the datasets evaluated for the execution of this project, selecting the one made by Cabrera [89], since it is the database in which the quality of data acquisition as well as the integrity of the data can be guaranteed, described in the same table.

As can be seen in table 6.1, the database consists of 55 subjects in an age range from 9 years to 50 years, which 33 are men, and 22 are women; within all subjects, five people have been diagnosed with ADHD or a similar disorder.

The selected database[89] was developed by performing three psychometric tests on a screen in the NEURO-INNOVA KIDS® [97] software in a controlled environment obtaining the trajectories using an eye tracker, Figure 6.2 shows the raw data obtained through the software and the eye tracker, which will be the data that will be preprocessed to get the images with trajectories showing the saccadic movements and fixations.

The software delivers two files, a TSV file and a Txt file containing the relevant information, which is the TSV file. Table 6.2 shows an extract with the values used to determine the trajectories.

Database authors	**Article name**	**Database characteristics**	**Access (URL)**
Cabrera [89]	A dataset on eye movement tracking during the resolution of neuropsychological tests on a screen	Total: 55 volunteers, 33 men, 22 women in an age range of 9 to 50.	Data in Brief [90]
Cheng [91]	Dissociable rhythmic mechanisms enhance memory for conscious and non-conscious perceptual contents	Total: 72 healthy young participants, 29 men, mean age: 20.56 years ± 0.41, 7 men, mean age: 22.17 years ± 1.32, 8 men, mean age: 19.17 years ± 0.26, 7 men, mean age: 19.89years ± 0.50, 7 men, mean age: 20.39 years ± 0.41.	Open Science Framework [92]
Selivanova [93]	Eye tracking to explore attendance in health-state descriptions	Total: 10 participants, no more information available.	Figshare [94]
Vortmann [95]	Multimodal eeg and eye tracking feature fusion approaches for attention classification in hybrid basis	Total: 51 students. Fifteen participants were excluded from further analyses because of excessive missing data(n=14) or technical problems (n = 1) during data acquisition. The final sample thus consisted of 36 subjects (24 female & 12 males), with an average age of 24 years ± 2.72.	Open Science Framework [96]

Table 6.1: Databases characteristics

timestamp	time	fix	state	rawx	rawy	avgx	avgy	psize
MSG	2021-01-26 1:	1070223302	IMAGENAME domino01.jpg					
42:39.3	1070223320	False	7	993.8597	521.7207	989.4279	545.1093	20.81495
MSG	2021-01-26 1:	1070223321	image online at 38140					
42:39.3	1070223337	False	7	991.2567	543.9078	990.4807	543.9888	20.28905
42:39.4	1070223370	False	7	987.1713	520.0859	991.9171	543.4952	20.58185
42:39.5	1070223503	False	7	0	0	0	0	20.4475
42:40.0	1070224003	False	7	270.5856	278.5	493.5139	355.2659	15.748
42:40.2	1070224186	False	7	240.8563	310.1756	303.2414	241.4729	13.88995
42:40.2	1070224203	False	7	211.5278	245.9273	300.7155	241.318	13.78985
42:40.3	1070224286	False	7	244.7892	244.7289	245.6798	234.1431	13.62625
42:40.5	1070224485	True	7	338.0976	241.4521	263.472	267.4751	13.83765
42:40.5	1070224536	True	7	232.4062	331.8799	262.7594	280.1548	14.38765
42:40.6	1070224569	True	7	236.6113	268.283	256.3541	283.4707	13.98765
42:40.8	1070224785	False	7	436.0139	426.6423	395.2192	411.2395	13.73065
42:40.9	1070224919	False	7	392.841	486.7181	430.002	439.7447	13.55205
42:41.0	1070224985	False	7	427.1746	415.3795	426.0832	447.7189	13.87775
42:41.1	1070225052	False	7	323.1325	484.6374	388.7195	421.4795	13.88295

Figure 6.2: Visualization of data obtained through the eye-tracker

Table 6.2 shows the most relevant values that were used to determine the trajectories, where Timestamp indicates the time in which it was recapitulated; AvgX and AvgY, is the average of both eyes in the Cartesian plane (X, Y), which serves to estimate where the subject was looking at.

Variable	**Description**
timestamp	Timestamp of the recording
Avgx	Average value of both eyes on the X-axis
Avgy	Average value of both eyes on the Y-axis

Table 6.2: Most Relevant TSV Values

The eye tracker used for testing and data acquisition was an EyeTribe, which, by scanning the eyes and face, can calculate the coordinates (x, y) position of the eyesight during the test, Table 6.3 shows the technical specifications of the eye tracker. Furthermore, the volunteer participants followed a protocol for the correct data collection, which consisted of sitting at a certain distance from the screen(60cm), following the indication; meanwhile, the EyeTribe was calibrated through the Software by detecting the eyes.

Specification	**Value**
Sample Rate	60 Hz
Sample Time	16.6667 ms
Fixation Threshold	1.5 degrees
Speed Threshold	35 degrees/second
Acceleration Threshold	9500 degrees/second^2

Table 6.3: Technical Specifications of the EyeTribe Eye Tracker

6.2.1 Psychometric Tests

Selecting an appropriate measurement instrument to determine individuals' attention levels is crucial. Validated and reliable assessment methods are required to ensure the activation of specific attentional states during testing. This thoroughness is critical to ensuring the integrity and applicability of the study's results.

These psychometric tests have been used to determine a person's level of attention due to their characteristics since there is a dependence on cognitive ability, more specifically on attention, for the ideal performance of these tests [98]. Although these

validations are of an indirect type due to subjectivity, they help to provide a preamble of what level of attention the person has.

In line with this requirement, the psychometric tests used in the selected database comprise a variety of tests adapted from different test models. These were chosen with particular attention since they are crucial to achieving the main objective of the study: to capture the visual trajectories and determine the level of attention of the volunteers. The meticulous selection of these tests reflects a dedication to ensuring that each study component effectively validates the chosen measurement method. Those tests are:

- **Domino test**: Is a test adapted from the D 48 test [99], developed by the psychologist Edgar Anstey in 1944, is a test of non-verbal intelligence, which means that it does not depend on language or cultural context; consists of a series of sequences of dominoes that have to complete the logical sequence of dominoes that are presented in different patterns, is a test that serves to evaluate the ability of reasoning and logic, its functionality in this project is that nonverbal tests are often used to measure the cognitive ability of people [100].

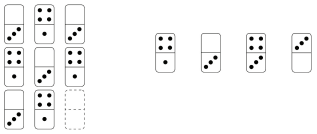
- **Unfold Cubes test:** It is a test that serves to measure the ability of people to form mental images [101]; it consists of an unfolded cube which must be mentally reconstructed and select the entire cube, it is a test of spatial visualization and abstract reasoning, without the need for language skills, meaning that it is a nonverbal test, which in this context as mentioned serves to determine the cognitive ability of people, focused on attention [100].

- **Figure Series test:** It is a test of inductive reasoning, used as its name suggests to measure the ability of inductive reasoning, which is to complete the series of figures to determine which figure would be the next to continue the sequence, it seeks to measure the ability to find patterns thereby facilitating to measure the cognitive ability of test subjects [102].

Figure 6.3 illustrates tasks from each test. The Domino test 6.3a shows a sequence of dominos with a missing piece that the subject needs to identify based on the pattern. In the Unfold Cubes test 6.3c, the figure depicts several views of an unfolded cube alongside a correctly folded cube to highlight the task of spatial reconstruction. Finally, for the Figure Series test 6.3b, the figure includes a sequence of shapes with one missing, asking the viewer to determine the following logical figure in the series, demonstrating how it assesses pattern recognition and inductive reasoning skills.

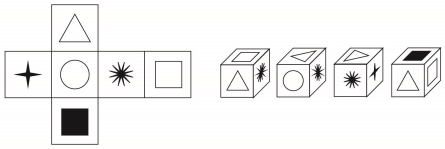
6.3 Preprocessing

Preprocessing is a critical step when training an artificial intelligence model because it is the previous stage in which the data will be passed as inputs depending on the model's characteristics. The goal is to optimize all the input data resources to obtain the desired accuracy. In the particular case of the project, two artificial intelligence



(a) Domino test

(b) Figure Series test



(c) Unfold Cubes test

Figure 6.3: Example of the psychometric tests used in the database to measure attention levels [89].

models had to be combined, and a specific preprocessing stage had to be carried out for each one. A Deep Learning model was used to extract features from the generated images, and these extracted features were used as input data for the selected machine learning models. Diagram 6.4 explains the steps to follow to preprocess the data.

6.3.1 Data cleaning and imputation

The first stage of preprocessing is data cleaning; this consists of determining what to do with missing values, outliers, and repeated values irrelevant to the project; this step is crucial to ensure the quality and efficiency of the data to be used since failure to remove the data can cause distortions of learning and therefore of the result [46].

When describing the database, the data that will be worked with are the AvgX and AvgY values, which have missing data and outliers; for the treatment of outliers, what was done was to establish limits on the two axes (X, Y), for the lower X limit that the data did not decrease from 0 since the screen does not have negative coordinates, The upper limit was set at 1920, since this is the number of pixels contained in the screen, a similar methodology was used for the Y limits, the lower limit is also zero since there can be no negative values. The upper limit value was set at 1080 since these are the physical characteristics of the screen.

Any value outside these ranges would be erased because it would produce a decompensation in the data when determining the trajectories or when making the data

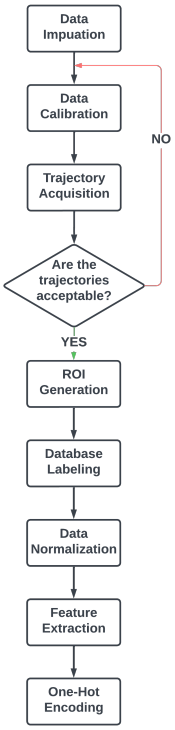


Figure 6.4: Preprocessing Diagram Flow

imputation. Table 6.4 shows an example of a test with out-of-range values, then a filter must be applied to remove those values.

Column	**Allowed range**	**Out-of-range values**
avgx	(0, 1950)	70
avgy	(0, 1080)	41

Table 6.4: Number of values outside the specified range per column.

The data shown in the table shows how it would be after applying the filter, ensuring that it works with values within the ranges established by the software and the screen.

Column	**Allowed range**	**Out-of-range values**
avgx	(0, 1950)	0
avgy	(0, 1080)	0

Table 6.5: Number of values after applying the filter.

On the other hand, missing data imputation is a stage of data preprocessing that involves replacing missing or null data using mathematical or statistical techniques, which allows statistical analysis to be more efficient in Machine Learning models [46].

It is of utmost importance that the data with which is going to be worked are primarily complete because otherwise, it could not guarantee the effectiveness with which the model learns; when reviewing the database, it was noticed that there are subjects to which, due to errors of the eye tracker, the subject saw off screen or failure in the software, the eye trajectories were not captured, consequently in the dataset are shown as null data, The subject saw out of the screen, or software failure did not capture the eye trajectories, therefore, in the dataset, they are shown as null data, and as in the case of this it is worked with trajectories it is of utmost importance that no value is missing because that will help us to determine both fixations and saccadic movements. That is why different techniques and methods of data imputation were explored to determine which one can best fit the nature of the data.

Data imputation techniques can range from simple methods, such as median or mode, to more advanced techniques that include data prediction techniques or statistical modeling techniques to estimate missing values using the remaining values in the dataset, thus ensuring that the imputed values make sense with the data already in the dataset [46, 103].

The technique used in the project is known as Cubic Splines, which is a mathematical method for data interpolation [104]; it consists of creating a smooth and constant function that passes through the given data, creating a segmented third-degree polynomial, which means that both the first and second derivatives are guaranteed to be continuous throughout the entire range. This technique was chosen because since the data are both temporal and follow a trajectory, an abrupt change in the data when using another technique could cause problems generating the trajectories.

In addition to those described above, cubic splines' main objective is to provide a smooth curve between the two points of the given segmentation, avoiding abrupt changes, which, as mentioned above, is what is trying to be avoided. This feature makes them ideal for the dataset since it is required that the missing data have an accurate and smooth representation of the existing data.

Linear Spline: To understand in depth the mathematical operation of the cubic model, it must be first understood the Linear Splines, which are extra similar but use straight line segments to connect both points of the given range; it is the simplest method within the family of splines, where each segment is a polynomial of the first degree [105, 106], despite that, this method guarantees that the polynomial passes through all the given points. However, this method's simplicity is an easy alternative, without too much computational expense, when the imputed points are not required to have smoothness between them.

A linear spline is formed by connecting a set of points (x_i, y_i) with $i = 1, 2, \ldots, n$. Each segment between two consecutive points (x_i, y_i) and (x_{i+1}, y_{i+1}) is defined by the equation of the straight line joining them:

$$y = m_i(x - x_i) + y_i \tag{18}$$

where the slope m_i is calculated as:

$$m_i = \frac{y_{i+1} - y_i}{x_{i+1} - x_i} \tag{19}$$

These segments ensure that the interpolated function is continuous over the entire interval but not necessarily smooth since the first derivatives at points x_i are not constant. The linear spline is suitable for data that vary linearly between successive observations.

Quadratic Spline: Quadratic Splines are second-degree polynomial functions used for data interpolation [107, 108]. The quadratic spline function in each interval (x_i, y_i) and (x_{i+1}, y_{i+1}) is defined as:

$$P_i(x) = a_i(x - x_i)^2 + b_i(x - x_i) + c_i \tag{20}$$

Where a_i, b_i, and c_i are coefficients that are determined to ensure that the function is continuous and that its first derivative is also constant at the junction points. This is accomplished by solving a system of equations that establishes equalities for the values and first derivatives of the polynomials at the points where they meet:

$$P_i(x_i) = y_i, \tag{21}$$
$$P_i(x_{i+1}) = y_{i+1}, \tag{22}$$
$$P_i'(x_{i+1}) = P_{i+1}'(x_{i+1}). \tag{23}$$

These conditions guarantee the smoothness and proper shape of the interpolated curve, avoiding discontinuities and abrupt slope changes at the junction points.

The difference between the cubic spline and the quadratic spline is using a lower-degree polynomial, which generates a smooth curve but tends to oscillate at the extremes of the functions. This could be a problem, as this is a case where such extreme changes should be avoided.

Cubic Spline: Cubic splines are advanced tools for data interpolation that use third-degree polynomials to ensure a smooth transition between data points[109, 104]. The equation defines each segment of the cubic spline:

$$P_i(x) = a_i(x - x_i)^3 + b_i(x - x_i)^2 + c_i(x - x_i) + d_i \tag{24}$$

Where a, b, c, and d are coefficients determined that the spline is continuous and differentiable up to the second derivative at the junction points.

Conditions of Continuity For the spline to be smooth, the following conditions are imposed:

- Continuity of the function at the spline points: $P_i(x_{i+1}) = P_{i+1}(x_{i+1})$.
- Continuity of the first derivative: $P_i'(x_{i+1}) = P_{i+1}'(x_{i+1})$.
- Continuity of the second derivative: $P_i''(x_{i+1}) = P_{i+1}''(x_{i+1})$.

Advantages of Cubic Splines Cubic splines are especially useful when data have variations in curvature. Compared to lower-degree splines, they allow for greater flexibility and accuracy in interpolation and better adapt to the natural shape of the data. As mentioned, the data are the trajectories, hence, it is a perfect way to blame the data that the quality of the data is maintained.

The table 6.6 shows the missing data values in the columns to be used after a data scan and deletion of the outliers, and the figure shows how the representation looks before and after the data imputation.

Column	**Missing values (NaN) / 0**
timestamp	0
avgx	77
avgy	77

Table 6.6: Number of missing values (NaN) and 0 values per column.

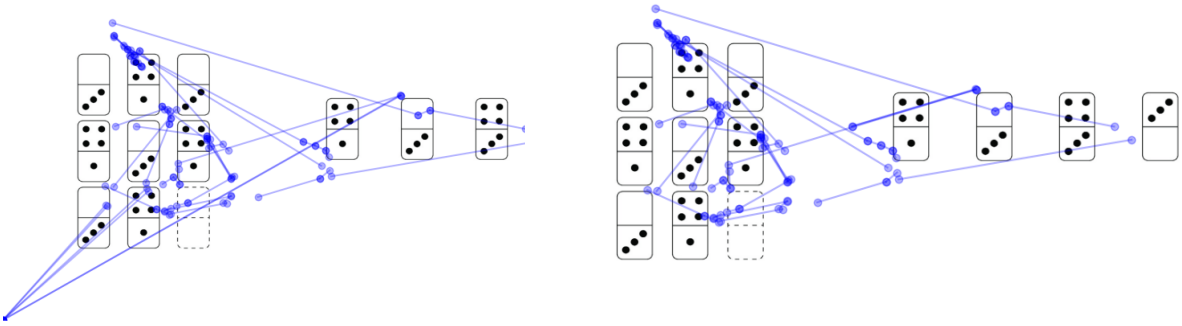
After data imputation, the values in the columns were updated as detailed in the table 6.7. The imputation method ensured that the original data's integrity and distribution were maintained as much as possible to minimize any bias introduced by missing values.

Column	**Missing values (NaN) / 0**
timestamp	0
avgx	0
avgy	0

Table 6.7: Number of missing values (NaN) and 0 values per column after data imputation.

Figure 6.5a shows how the missing data causes the trajectories to go to the origin, causing the model to have disturbances when learning from these features as they would be null values. In contrast, in figure 6.5b, it can be seen that the missing values

were imputed, ensuring that the nature of the trajectories is preserved without causing abrupt changes that prevent to continue maintaining the trend of the original data.



(a) Raw data

(b) Test after Cubic Splines imputation

Figure 6.5: Example of how the trajectories look like before and after having imputed the data using Cubic Splines

6.3.2 Data Calibration

Before determining the fixations and saccadic movements, a calibration adjustment has to be made because, during the data acquisition, it can be observed that, due to different factors, since the person has changed the position of the head, some aspects of the acquisition have changed, such as the distance between the head and the eyes. Therefore, some participants found an offset when reviewing the values in avgx and avgy.

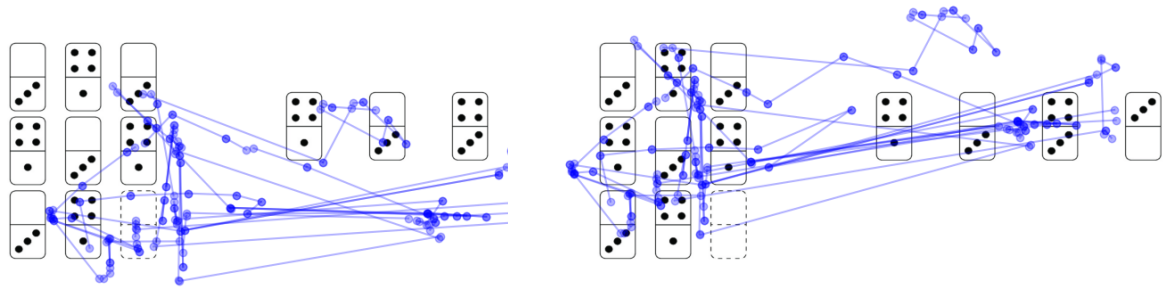
To perform this data calibration, an adaptation had to be made for each person who had an offset; a limit of 200px[11] was set for each value; to determine this calibration, the tests and subjects that exceeded this offset threshold had to be discarded since with such a large offset, it is not possible to guarantee the quality of the acquired test.

After discarding the subjects and tests that were not within the defined threshold without affecting the test, it was concluded that of the 55 participants, only 47 had a successful acquisition in most of the tests; it should be noted that of these 47, not all the information obtained can be used in all the tests since specific tests have a lag greater than the established limit.

Eleven tests were performed, but only four were selected because most of the subjects with a successful acquisition were obtained in those four; the table shows the subjects considered for each selected test, and the figure shows a visual representation from before and after the calibration of the data.

Test	Total participants
Domino 1	43
Domino 2	40
Unfold Cubes	42
Figure Series	47

Table 6.8: Total participants considered for each test.



(a) Raw data (b) Test after data calibration

Figure 6.6: Example of how tests change after data calibration

6.3.3 Obtaining fixations and saccadic movements

Once the raw data was worked to eliminate outliers or data out of the screen range, it is necessary to attribute the missing data to work with complete trajectories; the next step is to determine the fixations and saccadic movements; different approaches can be used to choose them, ranging from calculating the distance between point a and point b to determine the distance between each one and thus measure the area, as can be calculating the dispersion of the points and finally using a velocity threshold. The three approaches can be beneficial when working with trajectories; their main characteristics can be:

- **Area approach**: This approach, as mentioned, determines the distance between points a and b. Depending on the proximity generated between each point, calculating an area between these will generate an area that would be the fixation; if the distance between point and point is excessively far, it would be considered a saccadic motion.
- **Dispersion approach:** Its name indicates it, extra similar to the work in the

area, it uses the dispersion of the points to determine the fixations of the saccadic movements.

- **Velocity approach:** This approach considers that saccadic movements are fast movements compared to a fixation based on a static action; the movement velocity from point a to point b is determined to label a fixation or a saccadic movement.

The algorithm is known as Velocity-Threshold Identification (I-VT Algorithm) [110], which uses the velocity of the eye path to determine the fixation and is developed to analyze eye-tracking data efficiently. Its simplicity of development, efficiency, and ability to accurately segment eye-tracking data and why it was selected, and because a person's focus of attention represents the point at which attention is being paid, IV-T analysis helps us determine those fixations.

As mentioned, a fixation tends to be a slower and even static movement compared to a saccadic movement; the average velocity of a saccadic movement is in the range of 100°/s to 300°/s [111], this in angular velocity. The eye tracker used has an angular velocity parameter, which is easy to determine by using the minimum angular velocity to determine what a fixation is. The algorithm 1 developed and adapted is shown in the pseudocode, which describes how the function processes determine the fixations by returning a list of fixations.

The speed threshold used is 100°/s because a movement below that speed is no longer considered saccadic movement. The data entered into the algorithm are already preprocessed data; the process goes after having done all the data cleaning, data imputation, and calibration, and it has more efficient results.

Algorithm 1 Pseudocode of the I-VT Algorithm

```
Inputs: Eye-tracking data, velocity threshold v_thresh
Outputs: Fixations and saccades
procedure IVT(data, v_thresh)
    Initialize list of fixations and saccades
    for each sample i in data do
        Calculate velocity v_i between sample i and i + 1
        if v_i < v_thresh then
            Classify sample i as a fixation
        else
            Classify sample i as a saccade
        end if
    end for
    return list of fixations and saccades
end procedure
```

To determine the angular velocity using pixels, as already mentioned, the eye tracker has a parameter that indicates the minimum velocity to determine a fixation that could

be used as a reference. Another approach that can be used is the coordinates provided, although these are in pixels. Hence, pixels need to be converted to degrees to determine the velocity. Visual fovea needs to be calculated, which is the angle equivalent to the area of the retina where the luminous rays are focused, which, according to previous research, corresponds to 2° [112].

With the available information on the average distance between the participant and the monitor (60 cm) [89], it is possible to use a trigonometric ratio to calculate the size of the visual elements on the screen about the viewing angles. Considering that the visual angle forms a right triangle where the base b is the distance we want to calculate, and the height a corresponds to the distance from the observer to the monitor, we can apply the following formula based on the tangent of the visual angle:

$$b = \tan(\theta) \cdot a \tag{25}$$

In this equation, θ represents the visual angle in degrees, and a is the distance to the monitor in centimeters. Since the entire visual angle covers a particular area of the monitor, this formula calculates the length of the triangle's base in centimeters, corresponding to the amount of screen displayed. Furthermore, considering that the monitor has a physical width of 48.5 cm and a resolution of 1920 pixels, the equivalence between centimeters and pixels can be obtained:

$$1\,\text{cm} \approx 40\,\text{px} \tag{26}$$

Therefore, each degree of visual angle equals approximately 1 cm, translating to 40 pixels on the screen. This ratio allows one to calculate the size of the inspection area in pixels and the speed of the fixations in pixels per second, which is crucial for eye movement analysis. Knowing that 40 pixels equals 1 cm, and using the formula to determine the fixations as:

$$\text{Fixation} \approx v_{\text{thresh}} \times 40\,\frac{\text{pixels}}{\text{seconds}} \tag{27}$$

Substituting the defined threshold of 100px/s is equivalent; it gives us a value of 4000px/s. With this value, the fixations and saccadic movements can be determined.

Figure 6.7 shows an example of how the trajectories are visualized after having determined the fixations and saccadic movements, where the blue dots represent the saccadic movements, the red circles represent the fixations, and as can be seen, the longer the fixation time the more significant the area that is generated, this facilitates the understanding of the level of attention because it is expected to have more red dots than blue dots in the regions of interest.

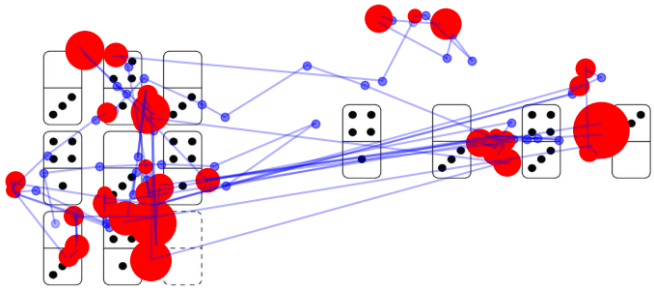


Figure 6.7: Fixation and Saccadic Movements obtained

6.3.4 Region of Interest (ROI) Generation

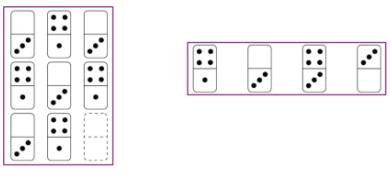
The region of interest (ROI) is a concept of utmost importance in the project and especially in matters related to eye-tracking tasks, as it refers to specific areas within an image, in this case within the test that people are performing, that one wishes to evaluate to understand how they are interacting with the visual content [113]. The ROI can be defined on critical elements that the researcher wishes to monitor precisely, such as images, texts, or, in the case of the project, within relevant regions where the subject is expected to pay attention when performing the test [114].

Establishing an ROI allows us to analyze better the patterns of fixations, which is the most relevant aspect since it differentiates one level of attention from another. These analyses help perform different studies and analyze people's behavior and environmental interaction. In behavioral studies, analyzing ROIs makes it possible to detect attention patterns and identify areas that generate greater interest and those that users ignore [115]. This information helps optimize the design of user interfaces, advertising, or even the layout of products in a physical store [116].

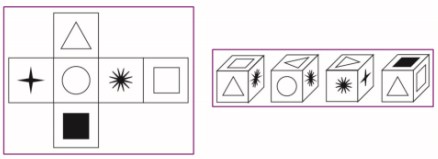
The ROIs help to understand the complexity of the activities because the more significant the complexity of the test, the greater the tendency to be fixated on it because it is not easy to perform. In addition, the inattentions in the ROI indicate that it does not sufficiently capture the person's attention for various reasons.

A relevant aspect is that they are customizable, which means that they can be defined according to the need and characteristics of the analysis; in this case 2, ROIs were defined, which focus on the section of the questions, which is where the subject is expected to have the highest number of fixations, and the second section which is the answers, in which the subject should have a lower number of fixations but still have in comparison with what is not within these two areas.

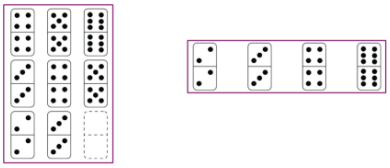
Figure 6.8 shows each test with a defined ROIs. The specific dimensions of each area must be established, a starting coordinate (X, Y) as well as a height and width, which must completely cover the question area and the answer area, to validate that it



(a) Domino 1 Test with ROI



(b) Cube Test with ROI



(c) Domino 2 Test with ROI

(d) Figures Test with ROI

Figure 6.8: Example of how the tests look like after the generation of AOIs

was correctly determined, a visual inspection of each test must be done.

The algorithm 2 exemplifies how the AOIs were defined by test, which draws a rectangle that encompasses each specific section; this will serve later to extract particular features and prevent our model from suffering any disturbance since most of the image is a white background, which would introduce considerable noise to our model.

6.3.5 Database Labeling

Once the fixations are obtained and the AOI is generated, the database is labeled; in the case of the selected database, the labeling was shared; this labeling is divided into three categories that are the ones that will be used in the classification project, which are:

- **Low attention:** In low attention, the person has trouble concentrating and tends to be easily distracted, and at a cognitive level, this leads to the person having more incredible difficulty in structuring visual attention tasks.
- **Medium attention:** A person with average attention can maintain considerable attention and concentration but still be susceptible to distractions and inattentions, yet still have structured patterns in visual tasks.
- **High attention:** High attention is characterized by intense and sustained concentration and ignoring external stimuli that may cause distractions. People's focus is usually within the areas of interest, and visual tasks do not represent a great difficulty when structuring patterns.

Algorithm 2 Process to generate AOIs

Input: AOIs' coordinates
Set left AOI as (x_1, y_1, w_1, h_1)
Set right AOI as (x_2, y_2, w_2, h_2)
for each image i from 1 to N **do**
    Load image i
    if image not found **then**
        Print error message
    end if
    Draw left AOI at (x_1, y_1) with width w_1 and height h_1
    Draw right AOI at (x_2, y_2) with width w_2 and height h_2
end for
Return Images with AOIs

According to the author [11, 89], these levels of attention were classified by the number of fixations and the average time they lasted. Still, only those within the area of interest were considered since they have a higher weight than other parameters measured and used in the test [117]; another factor is the total number of saccadic movements within the area of interest to have a complete picture of the trajectories.

Figure 6.9 shows an example of a test in which we can see the data necessary to obtain the level of attention; these are **SubjectId**, which helps us to determine which participant performed the test; **Fix_AOI_prom_duration**, which is the average time in seconds that each fixation lasted in the area of interest, **Fix_AOI**, which is the total number of fixations that were held within the area of interest, **Fix_AOI_questions**, which is the total number of fixations that were held only within the question section (left side of the tests) and **Fix_AOI_resp**, which is the complement of the above since it indicates the total number of fixations in the answer section (right side of the tests).

Other auxiliary parameters are optional but help to determine the level of attention more efficiently, **Transitions_AOI**, which are the number of transitions between one zone to another, that is, between the answers to the pre-questions. **Duracion_trayectoria** gives the total time in seconds of the duration of the test; **Fix_out**, this parameter is only a binary value that indicates if the person had any fixation outside the screen. This parameter also helps to determine if there was any inattention in the data acquisition process. **Dur_fix_ou** indicates the average time in seconds when the person looked outside the screen.

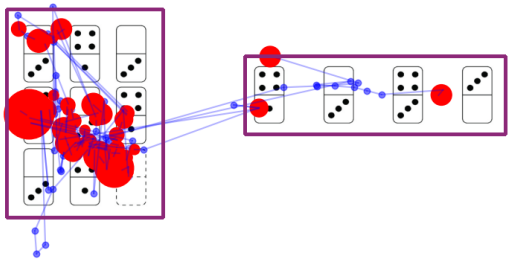
To consider a person with a high or medium level of attention, a visual inspection of both the test image and the calculated parameters is necessary since there must be a distribution between both areas of interest, and as mentioned, the higher the density of fixations, the more likely it is to have a high level of attention, another factor to take into account is that there must be at least one transition between both areas of interest since a participant with a medium/high level of attention must be able to inspect both sections in a controlled manner.

IdSujeto	Duracion_fix_prom	Fix_AOI	Fix_AOI_preguntas	Fix_AOI_resp	Transiciones_AOI	Duracion_trayectoria	Fix_out	Dur_fix_out	Num_fix	Nivel
1	0.058130081	78.04878049	62.5	37.5	5	3.65	5	0.106666667	41	1
3	0.083333333	93.84615385	62.29508197	37.70491803	22	7	2	0.125	65	1
4	0.085416667	100	54.16666667	45.83333333	10	2.75	0	0	24	2
5	0.241987179	100	65.38461538	34.61538462	23	13.75	0	0	52	2
6	0.087619048	97.14285714	50	50	11	3.883333333	0	0	35	2
7	0.063934426	73.7704918	66.66666667	33.33333333	11	6.95	3	0.094444444	61	1
8	0.139393939	100	59.09090909	40.90909091	13	3.666666667	0	0	22	2
9	0.159615385	92.30769231	52.08333333	47.91666667	25	9.35	0	0	52	2

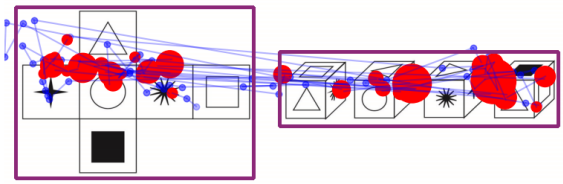
Figure 6.9: Parameters used for labeling the unfolded Cubes test, in which the level given to each subject can be observed, 0 = low attention level, 1 = medium attention level, and 2 = high attention level.

Finally, in a research study [118], it was shown that a person with a high level of attention tends to have similar search patterns; this means that they have an order at the time of inspection, resulting in similar at all tests, unlike someone with a low level of attention whose search pattern tends to be vastly dispersed on the screen in general, not only in the areas of interest.

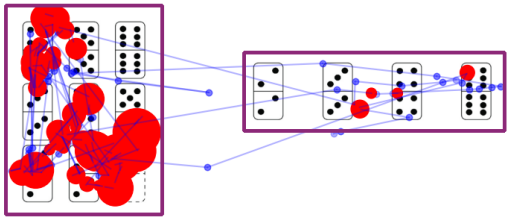
Figure 6.10 shows how a person with a high level of attention tends to keep his fixations within the areas of interest, in which several fixations are distributed between both areas of interest without going out of it.



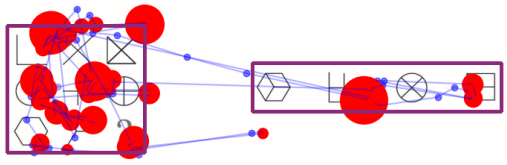
(a) High Attention Subject in Domino 1 test.



(b) High Attention Subject in Cubes test.



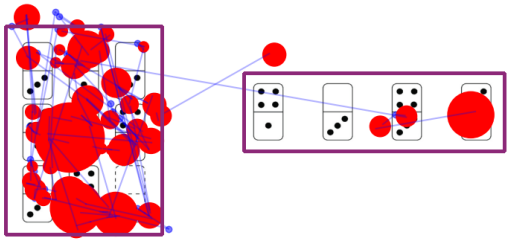
(c) High Attention Subject in Domino 2 test.



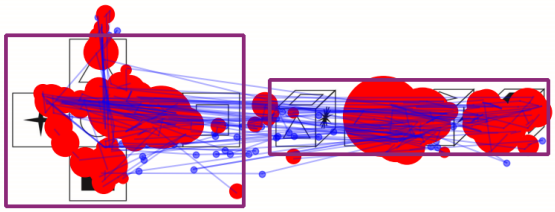
(d) High Attention Subject in Figures test.

Figure 6.10: High attention level tests

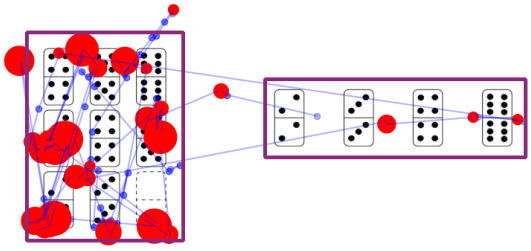
A person with a medium level of attention tends to have similarities with a person with a high level of attention; they have fixations in the areas of interest, and they have transitions distributed by both regions; the difference is that these people usually have distractions or inattentions that cause their attention to be different, although these inattentions are usually quite rare and last only a few seconds, figure 6.11 shows how people with a medium level of attention tend to have fixations outside the areas of interest.



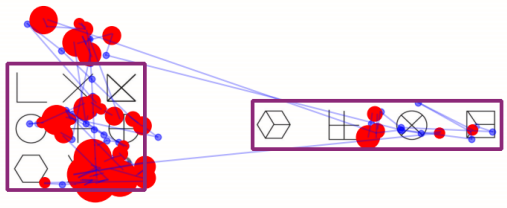
(a) Medium Attention Subject in Domino 1 test.



(b) Medium Attention Subject in Cubes test.



(c) Medium Attention Subject in Domino 2 test.



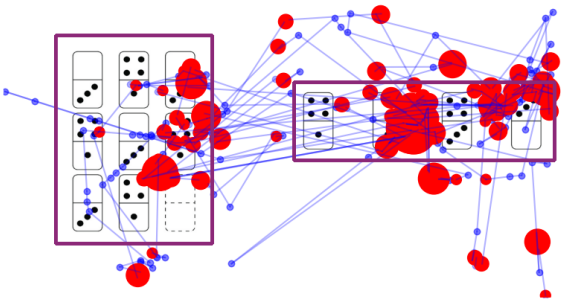
(d) Medium Attention Subject in Figures test.

Figure 6.11: Medium attention level tests

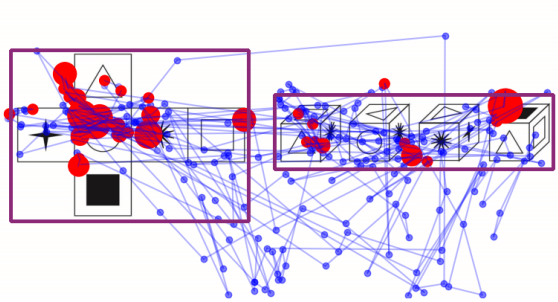
Compared to a person with a high level of attention, a person with low attention tends to inattention, which translates into fixations that do not focus on the relevant areas of interest or even drift away from the screen. This behavior reflects a significant difficulty in processing visual information effectively. At the cognitive level, the absence of a coherent search pattern suggests an inability to structure the visual task, affecting decision-making efficiency; these individuals have difficulty maintaining attention on a stimulus and may also have trouble reorienting themselves when distracted, affecting overall performance on complex tasks that rely on continuous visual attention.

Figure 6.12 shows that these patterns of inattention are extremely noticeable outside the areas of interest, thus it is easy to detect when a person has a low level of attention.

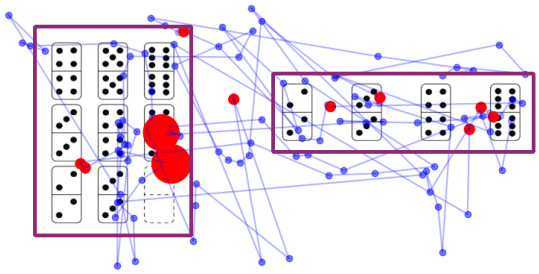
Table 6.9 shows the different tests with the total count of labels per test; as mentioned in previous sections, not all tests have the exact dimensions because specific tests had to be discarded. After all, they did not guarantee the quality of the test. Another notorious aspect that can be observed is class imbalance, an issue that will be worked on later to balance them.



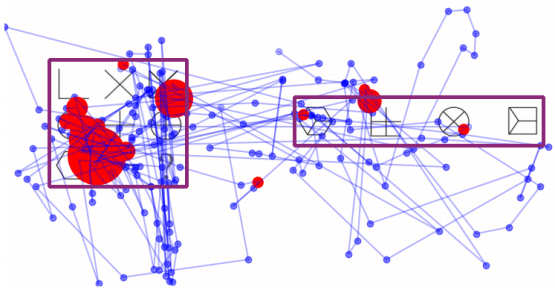
(a) Low Attention Subject in Domino 1 test.



(b) Low Attention Subject in Cubes test.



(c) Low Attention Subject in Domino 2 test.



(d) Low Attention Subject in Figures test.

Figure 6.12: Low attention level tests

Test	**Low Level**	**Medium Level**	**High Level**	**Total**
Domino 1	21	4	18	43
Domino 2	12	9	19	40
Unfolded Cubes	4	17	21	42
Figure Series	27	8	12	47

Table 6.9: Test Results by Attention Level Categories

6.3.6 Image resizing and Data normalization

Image resizing is an essential step in data preprocessing; it is a process in which the dimensions of an image are adjusted to a specific size, in this case, 224x224 pixels, to adapt to the inputs required by neural network models. This step is essential to ensure that all images have a uniform size, which facilitates training and inference in convolutional neural networks (CNN); for the case of the project, although a network will not be used as a classifier, it will be used to extract the features of the image as a previous step to a machine learning model. Among the main features used for resizing are:

- **Input consistency:** Convolutional neural networks, such as VGG16 or ResNet, require fixed-size inputs. If images are not resized, the model cannot process them uniformly. Resizing ensures that all images have the same dimensions, allowing them to be processed in batches and ensuring consistency during training. Common dimensions are 224x224 or 299x299 pixels, depending on the model [70].
- **Computational efficiency:** Image size can directly impact model performance. Larger images mean more pixels and more time during feature extraction. Standard size makes it possible to take advantage of parallel processing and optimize memory usage [119].
- **Adaptability to different data sets:** Resizing facilitates using pre-trained architectures (such as VGG16 or ResNet) with proprietary data. These architectures are designed to receive inputs with specific dimensions, adapting a set of images of different sizes to the required dimensions allows one to take advantage of transfer learning and obtain more efficient results on specific tasks [120]. Standard sizes also allow for maximizing parallel processing capacity and optimizing memory during training. This is essential in large models, which typically process hundreds or thousands of images per batch [121].

A VGG16 will be used for preprocessing in the project's specific case. Later, the reason for this particular network's use will be explained. A resizing of 224x244px was made with three color channels (RGB). The cv2 library [122], which has a function that suits the needs of the resizing, was used.

Data Normalization is a technique of scaling values in a specific range, usually [0,1]; for the particular case of the project to be working with images and use a CNN to extract features, it is necessary to perform normalization to ensure that the model can capture the features correctly, and work more efficiently [123], significant differences between pixels can negatively affect model training, especially in the first layers of the network [124]. The images in the project are 8-bit images, thus their range of values is usually in [0,255], and the following formula can present the normalization process:

$$x' = \frac{x}{255} \tag{28}$$

Where x' is the normalized value and x is the original pixel value. This process has to be performed on each image pixel to have a complete normalization.

This algorithm 3 is used for image preprocessing, specifically to prepare input data before feeding it into deep learning models for feature extraction.

Algorithm 3 Image Preprocessing Algorithm

```
Input: Image, image height, image width
Output: Preprocessed image
procedure PREPROCESSING(image, img_height, img_width)
    Resize the image to dimensions (img_width, img_height)
    Normalize the pixel values by dividing by 255.0
    return normalized image
end procedure
```

6.4 Data Augmentation

A fundamental step in the development of this project was the need to increase the data because, as seen in previous sections, the classes are unbalanced, and there is too little data to train a model and guarantee its accuracy. There is scarce data to train a model and guarantee its performance. Model and guarantee its performance. Therefore, a methodology was proposed to generate a data augmentation that would produce a unique and unrepeatable image, ensuring that the quality of the data generated synthetically was sufficiently diverse.

This proposed methodology shown in the algorithm 4 defines an objective function in which the number of images per class is established. Once this objective is defined, iterate through the original images of each class and randomly go through four previously defined image modification processes until the total number of images per class is reached, thus guaranteeing that only the original images are used as the basis for the generation of synthetic data.

The four defined processes consist of the translation of the image, which means the image is displaced horizontally or vertically on its center; this displacement is done randomly in a range ranging from [-10, 10] px, this assigned value, as well as the type of displacement is also random ensuring that the images do not follow a sequence but are unique; Zoom in the image, which as its name suggests makes a zoom in or out of the image, with a zoom factor ranging from [1. 0, 1.5] that is applied to both cases of the process. Image rotation: the image is rotated on its axis in a range of [-30°, 30°] that is randomly assigned to each image that uses this process; finally, the brightness of the image is modified by increasing or decreasing it, which is done randomly in the same way as the other processes.

Image 6.13 shows how synthetic images are generated. With these four processes explained, it is possible to create unique and unrepeatable images; in the same way, the

Algorithm 4 Image Augmentation and Class Balancing Algorithm

Input: Image dataset X, Labels Y, target size for balancing
Output: Augmented and balanced dataset
procedure AUGMENT_IMAGE(image)
    Apply a random translation to the image
    Apply a random zoom to the image
    Apply a random rotation to the image
    Apply a random color change to the image
    return Augmented image
end procedure
procedure BALANCE_CLASSES(X, Y, target_size)
    for each class in Y **do**
        Get all images for the current class
        Calculate the number of augmentations needed to reach target size
        for each augmentation required **do**
            Choose a random image from the class and apply a transformation to the image
            Add augmented image to the dataset
        end for
    end for
    return Augmented and balanced dataset
end procedure

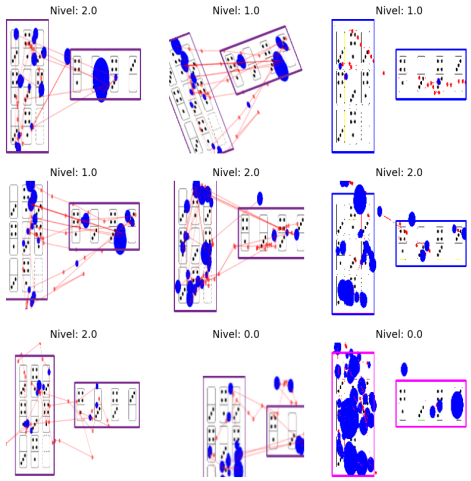


Figure 6.13: Augmented Images

number of unique images that can be generated from each original image is calculated.

This way, it is guaranteed that there are no repeated images. To determine the number of unique images, all the transformations and their parameters must be analyzed since, although it is randomly in specific ranges, the theoretical number of images is

limited per process. This is calculated as follows:

- **Translation:** Allows to move the image in the already explained range of [-10, 10] px; there is a displacement step of one pixel therefore that possible movements on the vertical axis are 21 and on the horizontal axis are 21, giving us a result of 441 unique images in this process.

- **Zoom:** It is scaled in a range of [1.0, 1.5]; considering that it is a continuous range of values, it could be determined that the number of values is infinite. Still, as it is required to have control of the possible images to generate, it is discretized to a step of 0.01, giving. As a result of only 51 unique images in the process, this range can be modified if the step is altered.

- **Rotation:** The range as mentioned is [-30°, 30°]; in the same way, if it is considered that it is a range of continuous values, the number of images is infinite, consequently a discretization was performed at steps of 1°, giving a result of 61 unique images.

- **Brightness:** It has a range of [0.8, 1.2] with a step of 0.01, which gives. As a result, 41 unique images.

As a result, the total number of images gives:

$$\textbf{Total Images} = 441 + 51 + 61 + 41 = 594 \tag{29}$$

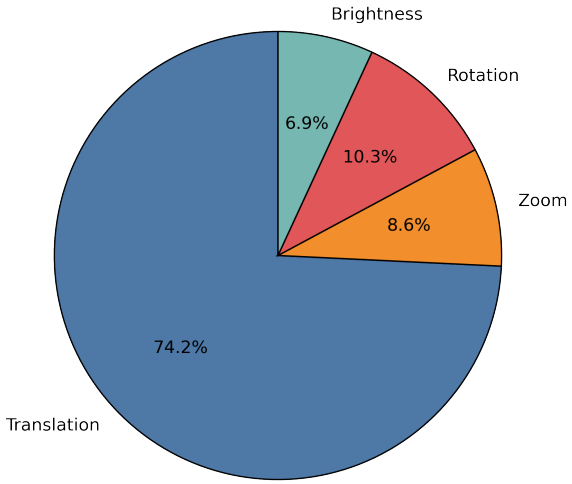


Figure 6.14: Percentage distribution of transformations

Figure 6.14 shows the total distribution of the percentage of how many images can be generated by each transformation, showing that translation is the process that would be applied in most of the images.

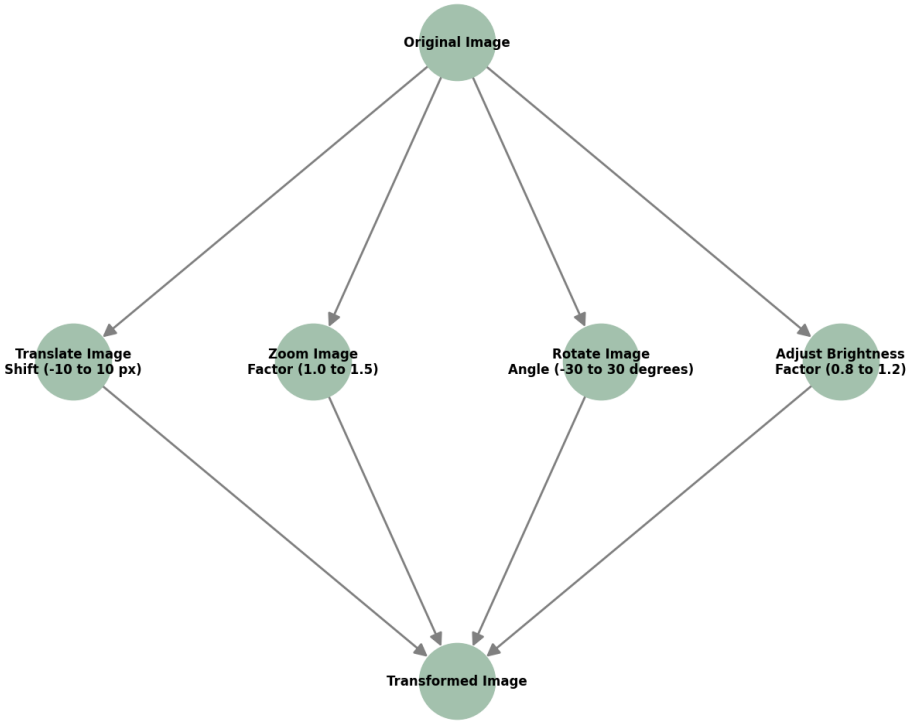


Figure 6.15: Data Augmentation random choice selection

Figure 6.15 explains the process for applying a transformation to each image. First, the number of images per class is counted. This number is compared with the previously defined target images to calculate the number of images per class needed to reach the target. That value iterates over each class until it is achieved.

6.4.1 Validation

Once all the images are generated, validating the image set is crucial. Various methodologies were explored for this task.

These methodologies were tested, but unfortunately, since the images were vastly similar from class to class and even within the same classes, a methodology was proposed that through dimensionality reduction techniques, such as the t-SNE technique, and diversity and affinity metrics, the variability of the images generated was checked, guaranteeing that they were unique and that two identical images of the same process were not generated.

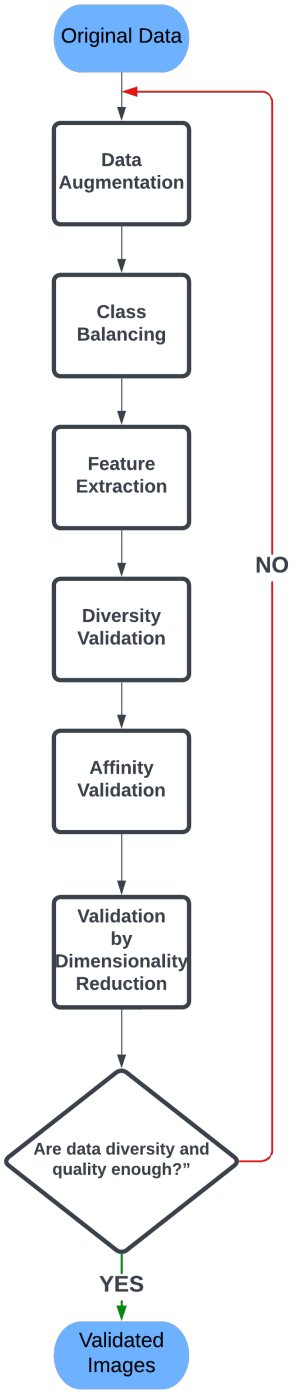


Figure 6.16: Methodology for data augmentation validation

Figure 6.16 shows the flow followed through the validation of the augmented images to ensure that the images are unique.

t-SNE The dimensionality reduction technique used, called t-SNE, is a technique generally used for the visualization of data in low dimensionality spaces (2 or 3 dimensions); it was beneficial because our vector of extracted features, being of too high dimensionality, would be challenging to analyze upon inspection, hence by reducing it to 2 dimensions it is easier to find patterns or trends when generating the images to be avoided.

If the visualized data shows an overlap, it indicates that the distribution of the data is inferior, furthermore more variability should be added to the images; on the other hand, if it shows a too abrupt separation of the data could be interpreted in two ways, which would be that the images are immensely different from each other which would be desired but in the case of the project to have similar images even from the original data is not desired because it would indicate that the generated images are not retaining most of the desired characteristics. Therefore, a balanced distribution is preferred, indicating that the generated images retain the primary characteristics of the original images but with enough variability for the model to learn from new features.

It should be noted that there are different dimensionality reduction techniques, but t-SNE was chosen because the data obtained do not follow a linear function. Hence, working not directly with the distances of the data but with the probabilities fits the project's needs.

Diversity Another important aspect of this validation is to determine that the data were generated successfully with the use of the diversity metric, which indicates the variability that exists between the original data and the synthetic data; this diversity gives us a relative numerical value, that is, the diversity of the original data has to be compared with the diversity of the generated data. This relative value depends on the set analyzed for the comparison to be relevant. In the case of the project, since the data are similar, the original and synthetic values should not vary much. Still, they shouldn't be remarkably identical since this would indicate that there is not enough variability.

Affinity Finally, it was necessary to validate that not only one class was taking all the diversity within the calculated value, consequently it was required to determine how much affinity there is per class, where the similarity of the images within the same category is determined, this could perhaps be considered as something counterproductive since it is intended to include as much diversity of images as possible, but the important thing is that these synthetically generated images do not lose the main characteristics of the class to which they belong. As with the validation of diversity, these are relative values that, unlike diversity, these values have to give as an average the result of diversity, indicating that the values are related.

6.5 Feature Extraction

As an essential step of the process, feature extraction is done just before being able to train the Artificial Intelligence models, which is crucial for training Machine Learning models in conjunction with computer vision. Where after having treated all the data in the previous steps and having obtained a final set of images ready for extraction, they are transformed into a set of manageable and representative features [123], in the context of our set of photos is focused on identifying patterns, such as edges, textures, shapes, and colors.

In feature extraction, one of the main goals is dimensionality reduction. Raw image data can be high-dimensional, increasing computational costs and potential overfitting. Feature extraction simplifies the data by identifying only the most relevant characteristics, ensuring the model processes essential information more efficiently [58]. There are two types of feature extraction:

- **Manual form:** It consists of an edge detection (Canny or Sobel operators) or a texture analysis (Gabor filters) that helps to identify critical visual patterns [125]. These methods are domain-specific and rely on prior knowledge of the task.
- **Automatic form:** CNNs automatically learn and extract features hierarchically through the convolutional layers. In the early layers, they capture simple structures such as edges, and in the deeper layers, they can detect more complex objects and patterns.

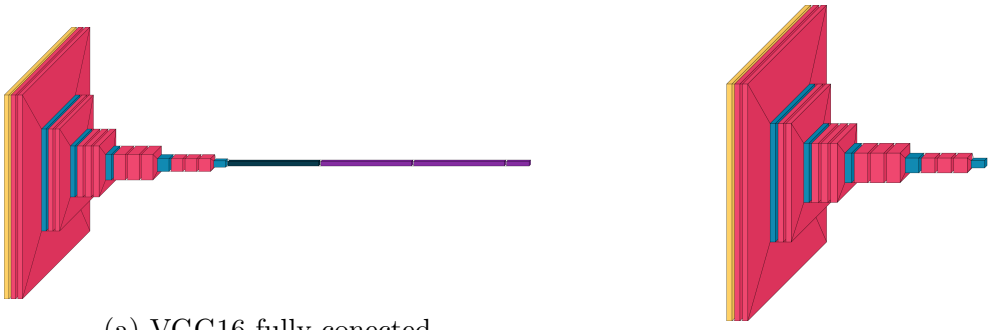
Once the features are extracted, they are represented as feature vectors, a numerical representation of the most essential characteristics of an image. These vectors are then passed to fully connected layers or other classifiers to complete the prediction task [70].

The project uses an automatic extraction based on a pre-trained CNN, named VGG16, which Simonyan and Zisserman developed [126], mainly used for recognition tasks; it is for this same attention to detail that it was chosen to use it as a feature extractor, and it adapts by freezing the deep layers remaining only with the convolutional layers that as explained automatically extract the essential features of the image.

VGG16 consists of 13 convolutional layers followed by five max-pooling layers. The 3x3 convolutional filters allow the network to capture fine-grained objects. At the same time, the pooling layers reduce the discretization of the feature maps, preserving the most critical information and reducing computational complexity. This hierarchical feature extraction process produces robust feature maps that can be used for various downstream tasks, such as object detection and image segmentation [127].

Figure 6.17 allows us to see the differences in the network structures. Figure 6.17a can observe that the fully connected network, which is the traditional architecture, has its flattening layer, which makes VGG16 become a complete network depending on the assigned task. Figure 6.17b shows how the network is used for feature extraction, which only uses the final layers.

At the end of the feature extraction through the CNN, vectors of size (N, 25089) were obtained, where N represents the number of images to work with and 25089 represents



(a) VGG16 fully conected

(b) VGG16 adapted for feature extraction

Figure 6.17: VGG16 architecture, adapted from [126].

the number of features per image, not working with masks on the image or different algorithms, the network focuses exclusively on the trajectories over the areas of interest. The image - shows a heat map in which the feature areas extracted from the images can be seen.

More networks were tested (EfficientNetB0, Resnet50), but extracted features, their execution time, in addition to computational overhead were considered as discard criteria. Among all the networks tested, VGG16 gave the best results in terms of the selection criteria.

6.6 Models Selection

This stage is essential in the research since it is the main activity proposed within the hypothesis, in which the model to be developed can obtain good results and good metrics. That is why the data preprocessing work was vital to ensure that the data passed to the model were correct since incorrect data generates inefficient algorithms.

It was decided to use ML models because the data being worked with and the detection of small patterns within the image are more suited to this type of model than to more robust models that require a much larger data bank than what is available. In addition, ML models are easy to develop and optimize.

In the selection process, it was considered to test as many Machine Learning models as possible, therefore, a tool called Pycaret[84] was used. This tool is a library developed for Python that focuses on testing the most popular models for the specific task selected. In the case of the project, it was classification. Within these models are the most popular ones, such as logistic regression, decision trees, random forests, XGBoost, Support Vector Machine (SVM), etc.

6.6.1 Models Development

Pycaret tool was used to set up the machine learning models, facilitating the development of the models in which only the correct dataset division had to be made. An essential feature of Pycaret is that the models are trained using K-Folds, where one

only selects the number of divisions with which it is required to work. This results in different metrics (Accuracy, Precision, F1 Score, Kappa, AUC, Recall), which help determine which model to use. All models were trained with their hyperparameters configured by default since the main idea of this stage is to determine which model has the best behavior toward the data.

6.6.2 Models training and testing

An important aspect is that there were two different stages of training and evaluation. In the first stage, all the tool's models were first tested without synthetic data to determine which models were best suited to the few data available. The second stage used synthetic data to compare again which models were best suited to the synthetic data, thus allowing the best model to be selected.

In the first training and evaluation with the original data, the data were divided into five folds since not having a lot of data is a quantity that allows the model to learn enough to recognize patterns of each class. Once the training and evaluation are finished, the data obtained are compiled and can be seen in the table (add a table in results), where it is shown which model was the best adapted for this stage; having unbalanced classes, it was necessary to include two more metrics that are specifically used for this type of situations, which are the Kappa and MCC metrics.

For the second stage of training and evaluation of the models, already including the synthetic data, the first thing that had to be done was an 80/20 division, in which the primary set of 2100 images in total was divided 80% (560 images per class) for training and 20% (140 per class) for validation. In turn, this 80% was divided again into five folds, then when validating the best-selected models, this 20% would serve as the best validation parameter. The models were trained with these five folds in which they were rotated through the same library that the model had as much data as possible; the table (table in results) shows the values obtained in the training and evaluation of the models through the K-Folds with the synthetic data.

6.6.3 Models validation

Unlike the training and evaluation, only a single validation stage was used since there was little original data, hence making an 80/20 division from the beginning was impossible. It was done with the synthetic data stage, which, as mentioned in the previous section, only 140 images per class, giving a total of 420 images; this stage of independent validation of the training and evaluation serves for the best models that are selected to serve as a kind of filter in which the best model is chosen since the models not having learned or seen the images previously are completely new data for these, it would be the best way to determine which is the one that best fits the data.

Table 7.4 shows the values of the four best-selected models, where the described metrics were considered a point of evaluation and selection.

6.6.4 Model Selection

After analyzing table 7.4, it is possible to determine the two models that better adapt to the validated data after one iteration: Logistic Regression and Light Gradient Boosting Machine. To ensure the correct model selection, it is necessary to train them individually using 10 K-Folds. This guarantees stability and, more critically, sustainable efficiency throughout the 10 individual trainings to support the selected model since it indicates that it behaves better.

Table 7.5 shows the results for the model selected. Once the model has been selected, the next step is optimizing it, finely adjusting the hyperparameters until it obtains the highest possible accuracy throughout the different validations.

6.7 Optimization

The Logistic Regression model was selected due to its adequate performance for the task to be developed in the project, with a quite acceptable accuracy; the next thing to do is to optimize these hyperparameters, which are the values that have to find the best combination to obtain the best result of the model. The table shows the values of the hyperparameters in the Logistic Regression model, where:

- **Penalty:** Is the penalty that will be applied to the model to prevent it from overfitting or not fitting the data correctly. In the case of the optimization, this value will not be moved and will remain at l2 because the l1 penalty tends to cause the model not to fit properly; hence, it is the reason only to use the l2 penalty.

- **Tol:** This is the tolerance of the stopping criteria; the default value is also used because a variation of this value can lead to the model not fitting correctly, resulting in even worse values.

- **C:** It is the inverse of the penalty value, thus this hyperparameter indicates how much value is given to the regularization; it means how much magnitude is given to the l2 values in this case, this being a especially important hyperparameter is considered to optimize it since it is a value that controls much in the model.

- **Random State:** It shuffles the data being worked with, but having similar classes changing by few characteristics, it will always find the global optimum of the solutions, this default value is sufficient.

- **Solver:** is the optimization algorithm used at the moment of minimizing the errors for an adequate adjustment in the training of the model; this parameter, having different algorithms to solve this optimization, the ones that are applicable for the case of the data and the task to be performed must be tested.

- **Max_iter:** It is the maximum number of iterations the solver algorithms have to converge the optimal values. Since this directly affects the solvers, it is another parameter that must be tested to determine its ideal value.

6.7.1 Grid search

It is a crucial technique because it maximizes the performance of the selected model's hyperparameters by focusing on an exhaustive search for the best values. A search space must be defined where all possible combinations between these chosen values will be tested, allowing the model to find the best values for each hyperparameter.

Therefore, taking again the hyperparameters selected in the previous section (C, solver, max_iter), the search space table is as follows:

Hyperparameter	**Values**
solver	lbfgs, newton-cg, saga
max_iter	500, 1000, 2000
C	0.0001, 10000

Table 6.10: Hyperparameters and their values

Table 6.10 shows the values to be tested during the optimization, explaining the reason for the values selected for these tests; in the case of **multi_class**, it simply serves to specify whether the model uses a binary or multi-class classification, in the case of the project it is the latter, but it must be specified in the model.

For the **solvers**, we selected those that are adapted to multi-class data and that are suitable for when there are a large number of characteristics and parameters to work with, unlike the other solvers that only work with binary problems; that is why it is essential to specify whether it is multi-class or binary.

The number of iterations (**max_iter**) was defined in a more defined range because when using large amounts of data, the default values of 100 are scarce. A more extensive range allows the model to find the optimization with all the algorithms.

Finally, perhaps the hyperparameter that most affects all the others is the **C** value, which, as previously explained, determines the importance of regularization in the model; this means how much the model will adapt to the penalties in each iteration. Small values that punish the model more are tested as large values that allow the model to reach convergence faster but are not always ideal.

6.8 Attention Model Classification

The final model is a hybrid of Machine Learning and Deep Learning. In this model, they are combined to extract the characteristics of the images using DL through a

VGG16. This vector of characteristics is the input of our ML model, which is a Logistic Regression. In this way, the model ensures that we can work with the images, assigning each subject its level of attention with a high degree of accuracy.

Table 7.5 shows the results obtained from the final hyperparameters after the optimization, which are the values with which the model will be trained for all the tests.

In addition, figure 7.5 shows the results of the confusion matrices, hence the next step is to train the model with different data orderings to ensure its efficiency.

For training and final validation, cross-validation with 10 folds was decided by repeating the experiment 25 times with the same hyperparameters. That is why, with all the 2100 images, a division of 10 folds was made. That is to say, 10 results are obtained per training, which can be observed in image 7.6, showing the model's behavior throughout the training.

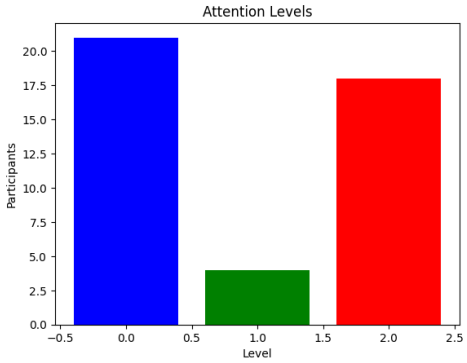
7 Results and Discussions

This section will present all the results obtained from this project, from class balancing and data augmentation, including the different results throughout the development of the models, to the selection of the final model with which the classification of the levels of care will be carried out.

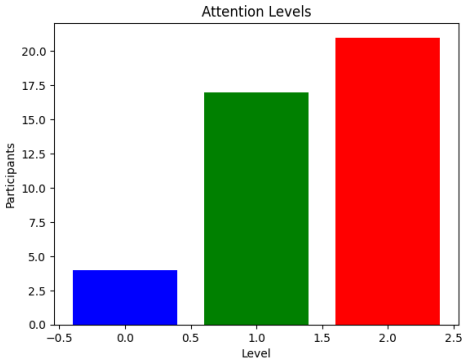
7.1 Class Balance

The table shows the values of the classes in each test. Also, figure 7.1 shows that they are entirely unbalanced without showing trends between them, therefore for correct training, it was necessary to perform class balancing.

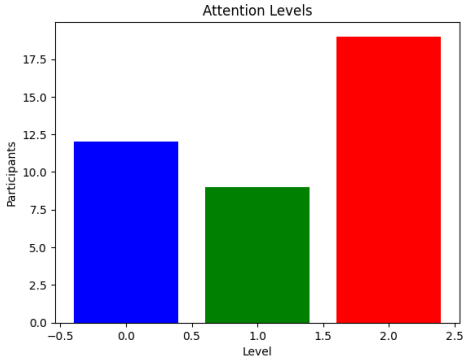
As explained in the methodology and in the algorithm 4, it was necessary to increase the data to perform this balancing, where the objective function was defined at 700 values per class. Through tests and experimentation, it was determined that after that figure, the gain in accuracy was completely null compared to the computational expense that had to be made to process them.



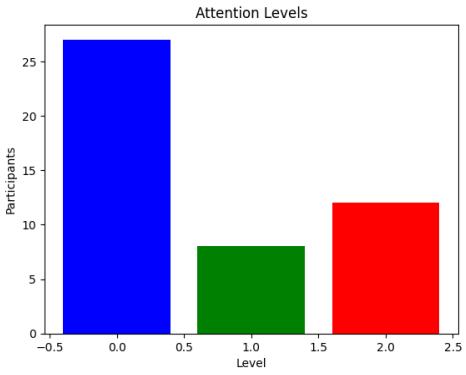
(a) Domino 1 Class Distribution



(b) Cubes Class Distribution



(c) Domino 2 Class Distribution



(d) Figure Series Class Distribution

Figure 7.1: Class Distribution per Test

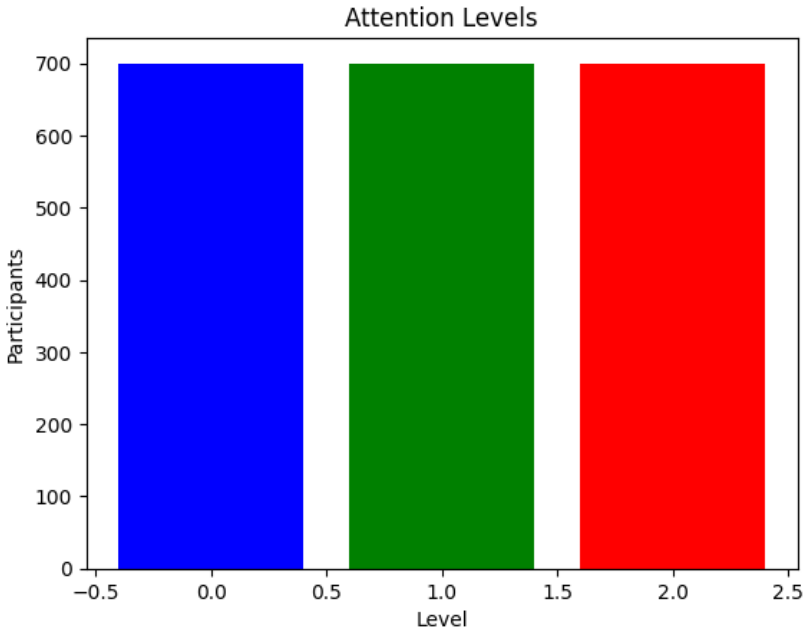


Figure 7.2: Classes after data balance

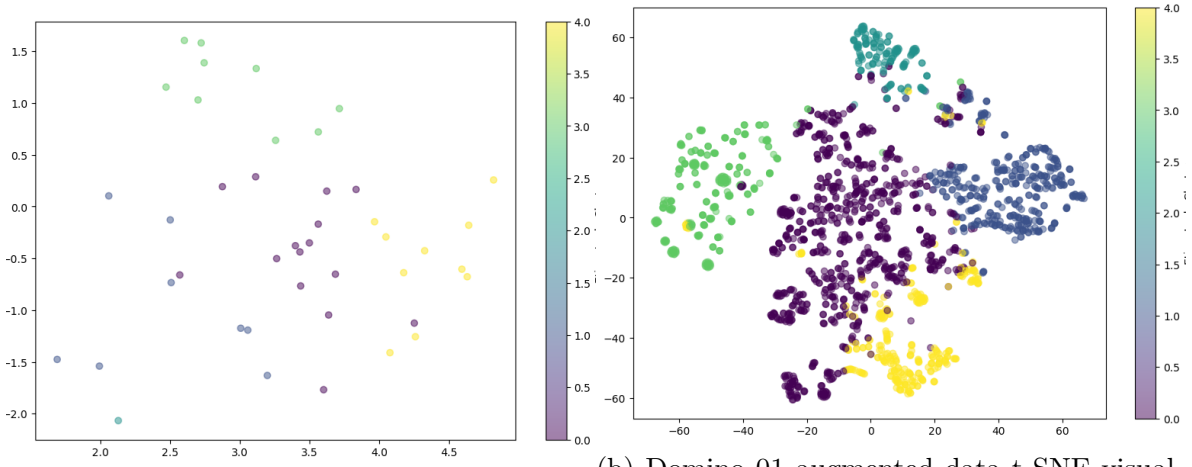
That is why figure 7.2 shows the uniform distribution of the classes after the balancing through the proposed data augmentation methodology, in which each class has the same number of images, allowing the model to have the correct training.

7.1.1 Data Augmentation Validation

To ensure that our class balancing by data augmentation is adequate, a methodology was developed to ensure that the synthetic data differs from the original data. Each test was validated independently since it is not possible to rely on the results of a single validation. This ensures robustness and certainty that the proposed methodology is functional in different types of tests and data.

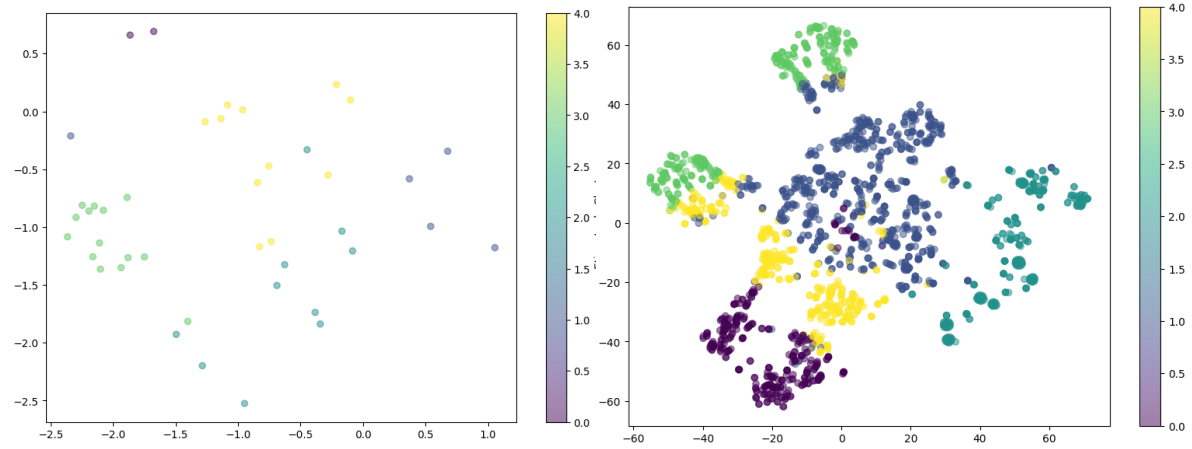
t-SNE When analyzing the distribution results by dimensionality reduction and comparing the original data with the synthetic data, it can be observed that the generation of this data was sufficiently enriching since there is quite adequate distribution in the plane.

Figure 7.3 shows the distribution of all the data generated, allowing a validation by visual inspection, this being the first step to ensure that the data augmentation was adequate. In which figures 7.3a and 7.3b show the comparative distribution between the original and the synthetic data of the domain values, figures 7.3c and 7.3d show that of the values of the displayed cubes and figures 7.3e and 7.3f show their respective values.



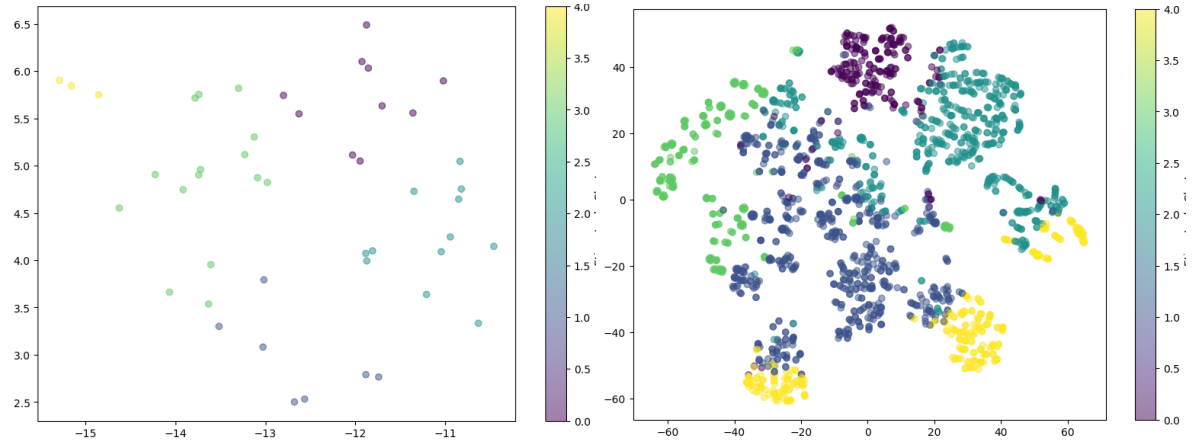
(a) Domino 1 original data t-SNE visualization

(b) Domino 01 augmented data t-SNE visualization



(c) Cubes original data t-SNE visualization

(d) Cubes augmented data t-SNE visualization



(e) Figure Series original data t-SNE visualization

(f) Figure Series augmented data t-SNE visualization

Figure 7.3: t-SNE visualization per test

Diversity Diversity metrics, which give the values of the variability of the images generated per test, are the next step in validating the data augmentation.

Table 7.1 shows the values obtained through the diversity calculation, observing a significant increase compared to the original values. This confirms the results obtained in figure 7.3, showing sufficient diversity in all data. It gives a numerical value as a parameter to estimate its efficiency when generating new data.

The average increase between the three tests was 9%, which indicates that the synthetic data are sufficient to balance the classes adequately and that the data are not equal.

Class	**Original Data**	**Data Augmentation**
Domino	16.9536	24.0319
Cubes	15.8990	26.4942
Figures	17.6986	27.1704

Table 7.1: Diversity of Classes

Affinity The last metric validates that the new images continue to preserve their class characteristics. This means they are not generating images utterly indistinguishable from the others in their category.

Like diversity, affinity represents a numerical value that allows us to estimate how acceptable our generation of synthetic data is, but with the difference that this is calculated by class since we want to evaluate the preservation of the primary characteristics of each class.

Table 7.2 shows the results obtained for each class. It can be observed that, despite the considerable variability of the images, these continue preserving their main characteristics since the increase in affinity was scarce.

This shows that the main characteristics of each image, despite being synthetically generated, are still preserved, thus allowing us to train the model with new images that still make sense of the category to which they are assigned.

Class	**Class 1**	**Class 2**	**Class 3**	**Class 1**	**Class 2**	**Class 3**
Domino	18.4720	15.7035	17.36037	24.1867	26.0103	26.7926
Cubes	16.2699	17.9823	19.8720	22.4138	26.5377	26.2795
Figures	16.1854	14.5449	18.3725	24.6234	26.0252	27.5354

Table 7.2: Affinity by Class

With this methodology of visual and numerical validations, ensuring that the images

with which the model will be trained have the necessary quality to guarantee adequate development is possible.

7.2 Experimental results

This section will report the results of the model development, including work done with the original data, with the augmented data, the optimization of the selected model, and finally, the validation of the model with the final hyperparameters.

7.2.1 Raw data training and testing

As a first step, different Machine Learning models had to be compared. In addition to the most common metrics (accuracy, precision, F1, Recall, etc.), the Kappa and MCC values had to be considered since, being unbalanced classes, metrics that evaluate the quality of the predictions without considering the volume of these had to be included.

As can be seen in table 7.3, the values of the standard evaluation metrics show that the four models could be sufficiently helpful when trying to classify the levels of attention; however, the Kappa and MCC metrics show that the LR and LightGBM models are the most sensitive when detecting trends in the patterns of each class.

It should be noted that 15 machine-learning models were compared with the original data. However, only the best models with an accuracy greater than 0.8 are reported since the other models could not adequately fit the data.

Models	**Accuracy**	**AUC**	**Recall**	**Precision**	**F1**	**Kappa**	**MCC**
LR	0.8683	0.963	0.86	0.8676	0.8666	0.8016	0.8027
LightGBM	0.8683	0.929	0.86	0.8967	0.8677	0.8017	0.8029
GBC	0.8587	0.903	0.8587	0.8589	0.8576	0.7873	0.7884
RC	0.8571	0.959	0.8571	0.8572	0.8561	0.7851	0.7860

Table 7.3: Models Evaluation Metrics

Analyzing the preliminary results, if an increase in data could not be performed, the final model would be the logistic regression model, therefore, it will be necessary to check through synthetic data how much this model can be adapted to new characteristics.

7.2.2 Synthetic data training and testing

Once the class balancing was done, all the models were tested again to compare which best fit the data now that the classes were uniform. Then, a decision was made about which model was adapting better to find patterns and tendencies alongside having good metrics.

In addition to having the classes balanced by increasing the data, the Kappa and MCC metrics are no longer necessary to calculate because they are only significant when the classes are unbalanced.

As can be seen in table 7.4, even though with the synthetic data, new models performed better than those previously reported, it can be determined that the logistic regression model continues to have the best results compared to the other models.

Models	**Accuracy**	**AUC**	**Recall**	**Precision**	**F1**
Logistic Regression	94.29	98.248	94.2875	94.3733	94.2665
Ridge Classifier	93.42	98.53	93.42	93.47	93.40
Gradient Boosting Classifier	92.52	90.3	92.6	93.56	92.48
Extra Trees Classifier	93.56	98.45	93.58	93.56	93.53

Table 7.4: Models Evaluation Metrics

7.2.3 Model Validation

After experimenting and developing the appropriate models for the project, the next step is to select the model. After analyzing the different training stages, the logistic regression model was chosen.

It is necessary to mention that all the models were developed through Pycaret[84], thus the individual validation of the model was developed without the use of this tool, which will give us a comparison of the values obtained, allowing us to analyze whether the results are appropriate or not, using the same experimental parameters, five-folds, with the predetermined hyperparameters.

The confusion matrix is a good way to visually review the results and inspect the predictions made by each class.

Figure 7.4 shows the confusion matrix of the randomly chosen test to determine the model's functionality before optimizing it, in which it can be seen that the model has a good sensitivity to detect the primary characteristics of each class.

7.2.4 Optimization

Optimization is essential in developing artificial intelligence models because it involves finding the best possible combination of hyperparameters.

The technique shown in the methodology section was a Grid Search, in which different combinations of hyperparameter values were tested; table 6.10 shows the range of proposed values.

The Grid Search was performed by K-folds using K=10 to test all possible combinations, resulting in a total of 900 possible combinations, taking 12 hours of execution time using ColabPro's A100 graphics card. Resulting in the following values:

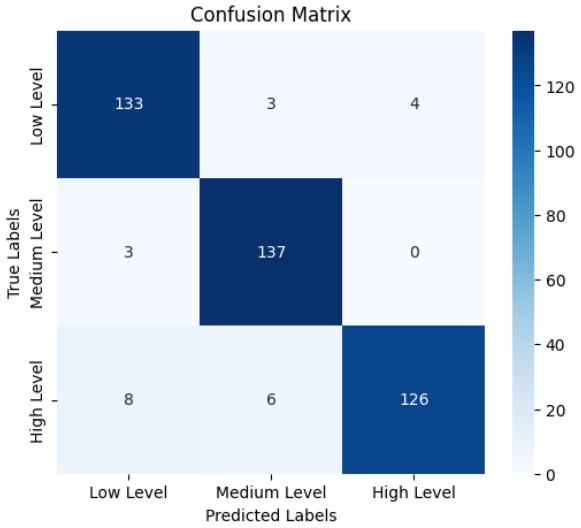


Figure 7.4: Confusion Matrix for model validation

- **C:** 2.782559402207126
- **max_iter:** 500
- **solver:** newton-cg

Then, as a next step, the model should be tested with the optimized hyperparameters using the same test performed in the previous step to compare the new values against the old ones.

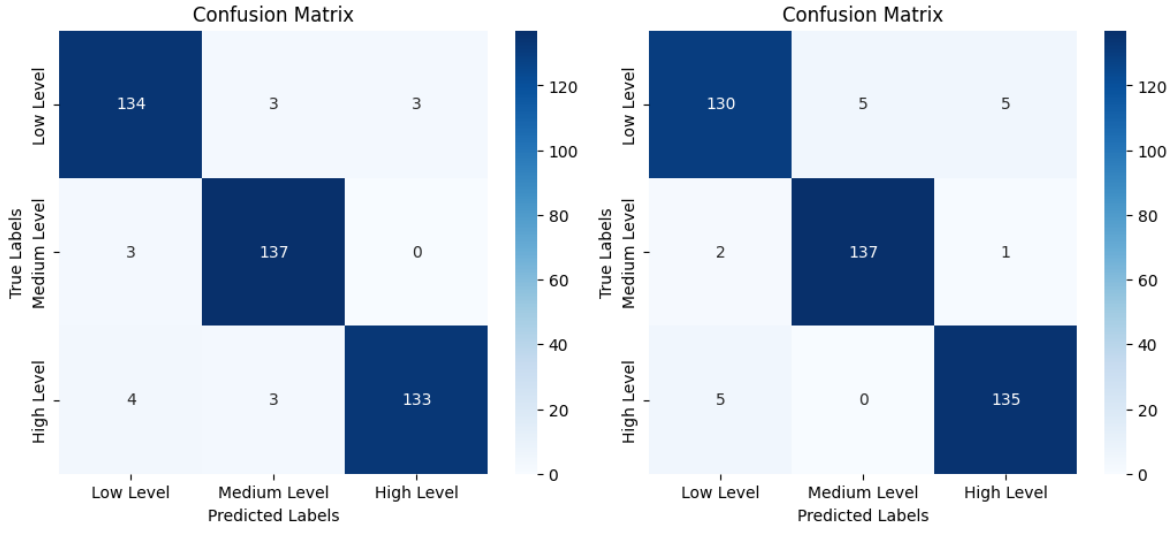
Comparing the results shown in table 7.4 with those of table 7.5, it can be determined that optimizing hyperparameters was successful since the model's evaluation metrics increased.

Model	Accuracy	AUC	Recall	Precision	F1 Score
Logistic Regression	95.85	99.2483	95.7285	95.761	95.6589

Table 7.5: Model Evaluation Metrics

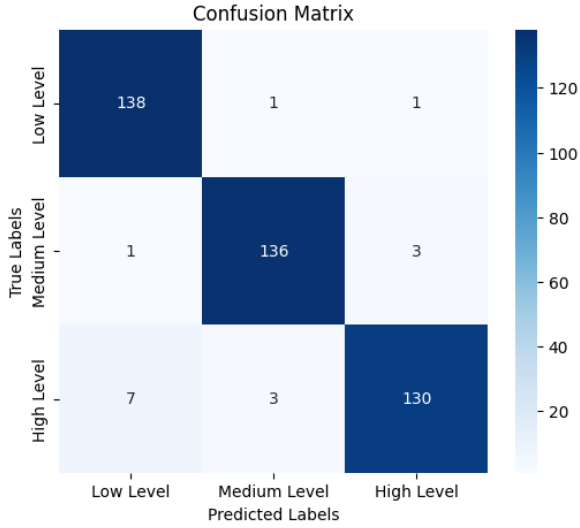
Figure 7.5 shows the results obtained when testing the model with the four different tests, observing that good results are inferring that the optimization was adequate then that the model could adequately detect the patterns and characteristics of each test.

Compared with the confusion matrix in the previous section, the confusion matrix is more sensitive to the correct classification, decreasing the number of false predictions in each class, where there are tests that even have zero errors in the prediction of the class, being this an outstanding result taking into account the similarity of the classes.

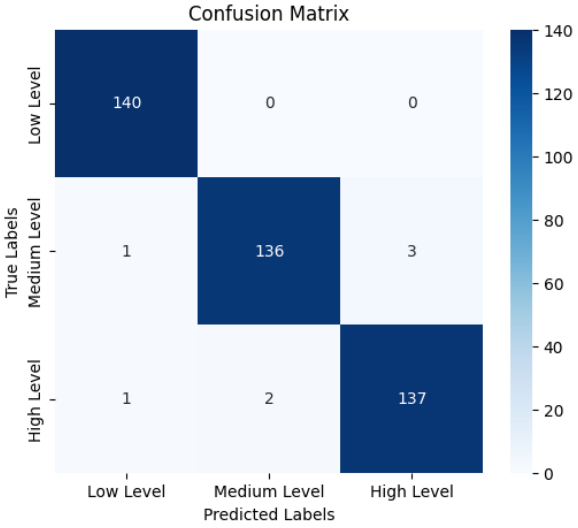


(a) Domino 1 Confusion Matrix

(b) Cubes Confusion Matrix



(c) Domino 2 Confusion Matrix



(d) Figure Series Confusion Matrix

Figure 7.5: Confusion Matrices per Test

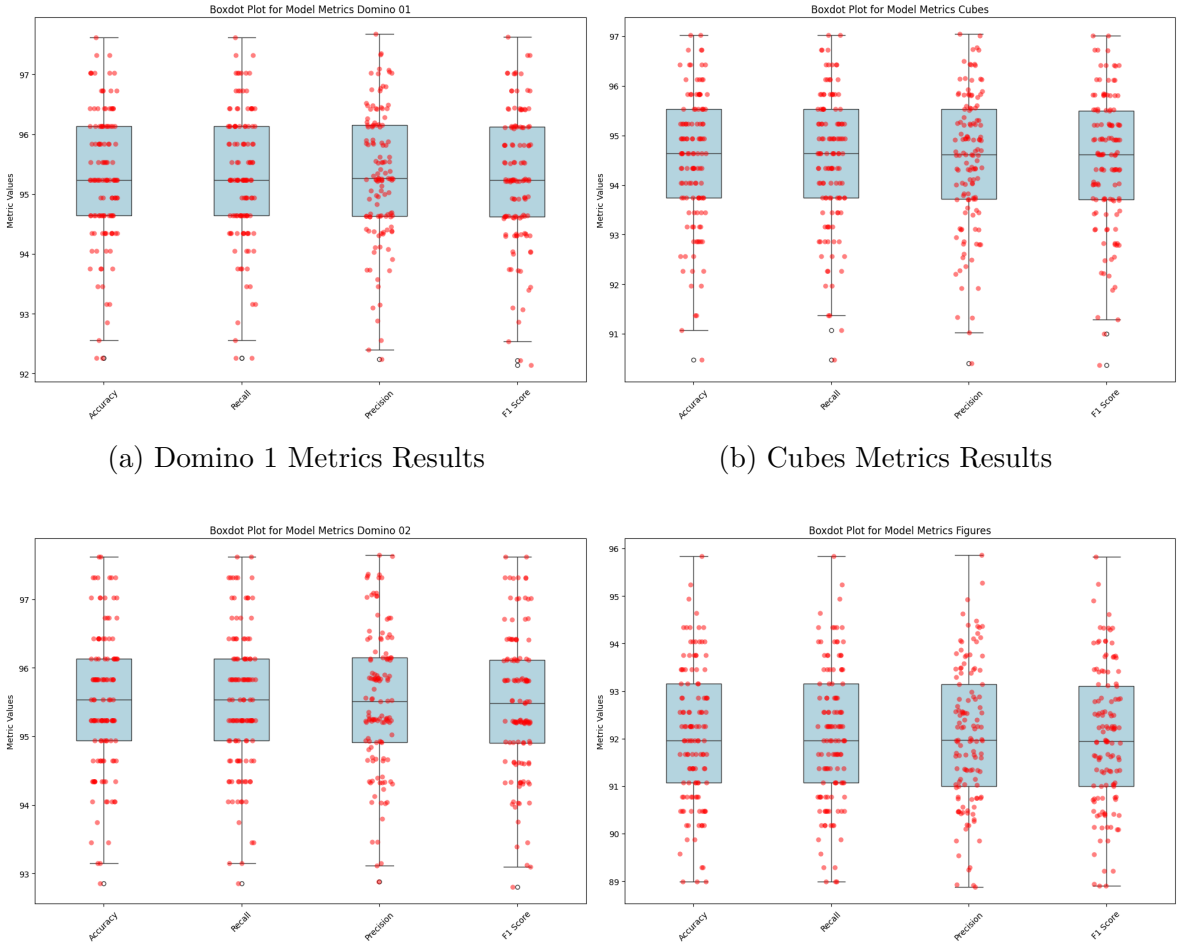
7.3 Optimized Model

As a final stage, once the optimization has been completed, it is vital to train the model to estimate that the evaluation metrics for each test are acceptable and are not just a matter of singular values; in other words, they have only been presented at that moment and will not return as a result.

For the final development of the model, it was decided to work with the 2100 data, hence an initial division of the data set was not made. A K-Folds cross-validation was

performed, using 10 folds for each training, with a total of 25 trainings, consequently, that the behavioral trend of our model can be adequately estimated.

The results of each test are presented in a box plot, which is a combination of the traditional box plot, which estimates the course of the model's behavior in each test, and a dot plot, which observes the results of each training. Thus, it can be visually inspected that each training was different from the previous one.



(a) Domino 1 Metrics Results

(b) Cubes Metrics Results

(c) Domino 2 Metrics Results

(d) Figure Series Metrics Results

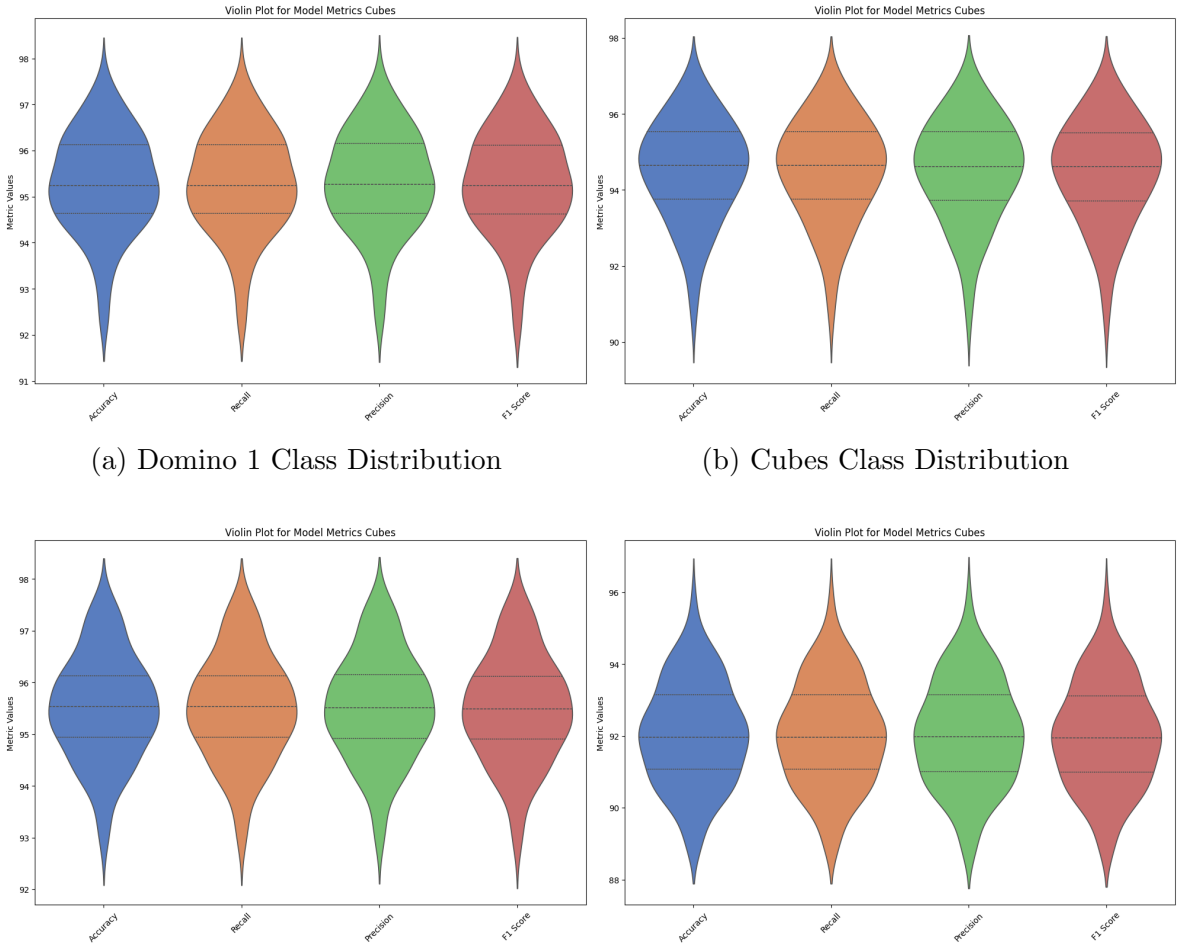
Figure 7.6: BoxDots Plot results per Test

From figure 7.6a, it can be determined that the average results for test 1 are between 95-96%, these being excellent results. Now, analyzing figure 7.6b, it can be seen that it has a performance below 95%, which, although it is not a bad result, shows that test 3 has lower results than test 1; on the other hand, the results of test 2 in figure 7.6c indicate that this is the test in which the model had better metrics within all the others being all highly close to 96%. In comparison, test 4, shown in figure 7.6d, is the one that had worse results than all the others, with an average trend of 92% in its metrics.

Table 7.6 shows the average values for each test after completing all the training sessions.

Test	Accuracy	Recall	Precision	F1 Score
Domino 01	95.28571428571	95.28571428571	95.31706625995	95.2706986537
Domino 02	95.49285714285	95.49285714285	95.51383369003	95.4739114020
Cubes	94.50476190476	94.50476190476	94.51063858946	94.478482082
Figures	92.04285714285	92.04285714285	92.06525334253	92.0055405090

Table 7.6: Evaluation Metrics for Different Tests



(a) Domino 1 Class Distribution

(b) Cubes Class Distribution

(c) Domino 2 Class Distribution

(d) Figure Series Class Distribution

Figure 7.7: Class Distribution per Test

Another way to visualize the results is employing a violin plot, in which the box

plot and a histogram are combined, allowing to observe in a better way the distribution of the data, where the shape of the "violin" is how the results are being distributed, the wider the section, the greater the concentration of data in that area, also in the central part it can be seen the box with the representation of the quartiles in a similar way to the box plot, and the bars denote the range of values from the minimum to the maximum.

Figure 7.7 shows the different results of the four tests, showing the trends, ranges, and distribution of results, giving an idea of how the model behaves. It is possible to see in figure 7.7c that it is the one with the better distribution, and the conclusion is the test that the model adapts the most.

8 Conclusions

In this project, different types of methodologies were proposed to achieve the main objective, which is to develop a model capable of classifying levels of attention; within the proposed methodologies is the data augmentation that guarantees to reach a minimum set of data per target class, making more practical how the balancing of classes is performed without the concern of generating repeated or equal images as long as the limit of possible unique images to be generated for each original image is not exceeded. This helps this research and could be replicated in other areas, allowing us to overcome the disadvantage of having limited databases.

The second methodology proposed was the validation of data augmentation when images or classes are extremely similar, using dimensionality reduction techniques and diversity and affinity metrics.

The development of an artificial intelligence model combining the use of both branches, such as Machine Learning and Deep Learning, has potential in attention level classification, achieving an average accuracy above 95%, exceeding 90% in all tests.

The methodology followed in the project presents several advantages in comparison with past research, which only focused on working with the original data. In the research developed, synthetic data were generated, in addition to the combination of two models extracting the characteristics of the images by means of DL and classifying using ML models.

The model represents the reliability that was proposed in the hypothesis because traditional methods, besides being invasive, presented a reliable alternative when developing this project due to their high accuracy and good precision, in addition to their sensitivity to adequately distinguish the classes.

The comparison of models was a turning point because this resulted in the final selection of the model to be developed, which was the project's main objective. If an inadequate model was chosen, it would present failures and generate a deficient model.

Another essential aspect that stands out in this research is the improvement of all the past research because they all report the accuracy metrics; comparing the metrics obtained in this project with other projects, it can be highlighted that all the research has been improved, being able to observe this comparative in the table. In addition, comparing the research with the previous direct work that was developed using the same database, it is demonstrated that using a hybrid model optimizing the hyperparameters of this one, a more robust model is generated than the one previously done.

Returning to the established objectives, the selected database was of the utmost importance because research can stagnate without an adequate database. When the database was chosen, different characteristics had to be evaluated, as well as the quality of the data, and one of the main aspects to consider was that it had to be public.

The final model demonstrated not only accuracy but also sensitivity and precision, as well as a great capacity for adaptation when presented with new and synthetic data, given it is possible to ensure that this work demonstrates that by using eye trajectories and artificial intelligence, it is possible to have a correct classification of the levels

of attention, making it an important tool for the help of health experts, joining AI with medicine allowing to generate better, more accurate diagnoses, not only focusing on attention disorders, but cognitive disorders in general due to the similarity of the conditions.

8.1 Future Work

One of the first possible future works is the acquisition of more data for model training, the integration of more tests to be able to have more comparisons between investigations, by acquiring more data it is possible to implement more robust models that cannot be developed with a limited data bank.

Generate links with public and/or private hospitals, in addition to having medical validation with specialists in the area; it is possible to have access to people with a diagnosed condition in order to have a point of comparison. Develop a system that not only uses eye trajectory data, but also generates an agent that accepts documents, audios, videos, etc., that provides a real-time classification.

Another aspect is to implement the model within the NEURO-INNOVA KIDS® Software, in which results could be given with greater precision, in addition to allowing learning with new data and generating a more robust model. In addition to all this, it is possible to have the validation of a person from the medical area to guarantee the sensitivity and accuracy of the results obtained by the model.

Bibliographic references

[1] F. P. Maragall, "Las capacidades cognitivas en el día a día," hablemos del alzheimer," 2021.

[2] I. Pascual-Castroviejo, "Trastornos por déficit de atención e hiperactividad (tdah)," *Asociación Española de Pediatría y Sociedad Española de Neurología Pediátrica. Protocolos de Neurología*, vol. 12, pp. 140–150, 2008.

[3] Y. Abdelrahman, A. A. Khan, J. Newn, E. Velloso, S. A. Safwat, J. Bailey, A. Bulling, F. Vetere, and A. Schmidt, "Classifying attention types with thermal imaging and eye tracking," *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 3, no. 3, pp. 1–27, 2019.

[4] F. J. R. Muñoz, "Aspectos explicativos de comorbilidad en los tgd, el síndrome de asperger y el tdah: estado de la cuestión," *Revista Chilena de Neuropsicología*, vol. 4, no. 1, pp. 12–19, 2009.

[5] S. P. Mansilla, "Enfermedades neurológicas y problemas de atención," *Acta Neurológica Colombiana*, vol. 22, no. 2, pp. 190–194, 2006.

[6] M. G. D. L. H. Vásquez and M. I. R. Hernández, "Relación de los movimientos oculares sacádicos y la comprensión lectora con el déficit de atención e hiperactividad (tdah)," *Inclusión y Desarrollo*, vol. 6, no. 1, pp. 137–149, 2019.

[7] J. Zhao, M. Wu, L. Zhou, X. Wang, and J. Jia, "Cognitive psychology-based artificial intelligence review," *Frontiers in neuroscience*, vol. 16, p. 1024316, 2022.

[8] E. A. Duéñez-Guzmán, S. Sadedin, J. X. Wang, K. R. McKee, and J. Z. Leibo, "A social path to human-like artificial intelligence," *Nature machine intelligence*, vol. 5, no. 11, pp. 1181–1188, 2023.

[9] A. Pachón, "Cognición artificial: delegación de inteligencia en la era digital," *ASRI: Arte y sociedad. Revista de investigación*, no. 17, p. 7, 2019.

[10] P. Kaushik, A. Moye, M. v. Vugt, and P. P. Roy, "Decoding the cognitive states of attention and distraction in a real-life setting using eeg," *Scientific Reports*, vol. 12, no. 1, p. 20649, 2022.

[11] M. G. Bedolla-Ibarra, M. D. C. Cabrera-Hernandez, M. A. Aceves-Fernández, and S. Tovar-Arriaga, "Classification of attention levels using a random forest algorithm optimized with particle swarm optimization," *Evolving Systems*, vol. 13, no. 5, pp. 687–702, 2022.

[12] M. Abercrombie, "Perception and communication," *Education+ Training*, vol. 8, no. 6, pp. 264–269, 1966.

[13] A. Belle, R. H. Hargraves, K. Najarian *et al.*, "An automated optimal engagement and attention detection system using electrocardiogram," *Computational and mathematical methods in medicine*, vol. 2012, 2012.

[14] M. M. Sohlberg and C. A. Mateer, "Effectiveness of an attention-training program," *Journal of clinical and experimental neuropsychology*, vol. 9, no. 2, pp. 117–130, 1987.

[15] K. Yoo, M. D. Rosenberg, Y. H. Kwon, Q. Lin, E. W. Avery, D. Sheinost, R. T. Constable, and M. M. Chun, "A brain-based general measure of attention," *Nature human behaviour*, vol. 6, no. 6, pp. 782–795, 2022.

[16] A. F. B. Díaz *et al.*, "Clasificación de estados de atención visual con métricas de seguimiento ocular mediante técnicas de aprendizaje profundo," *Repositorio UAQ*, 2023.

[17] M. Samiei and J. J. Clark, "Predicting visual attention and distraction during visual search using convolutional neural networks," *arXiv preprint arXiv:2210.15093*, 2022.

[18] A. Hazan, Y. Harel, and R. Meir, "Learning an attention model in an artificial visual system," in *2016 IEEE International Conference on the Science of Electrical Engineering (ICSEE)*. IEEE, 2016, pp. 1–5.

[19] G.-E. C. Alberto, A.-F. M. Antonio, C.-H. Carmen, and T.-A. Saúl, "Clasificación de niveles de atención con datos de desempeño y de seguimiento ocular durante la prueba adem," *La Mecatrónica en México*, 2024.

[20] J. E. Hoffman and B. Subramaniam, "The role of visual attention in saccadic eye movements," *Perception & psychophysics*, vol. 57, no. 6, pp. 787–795, 1995.

[21] O. T.-C. Chen, P.-C. Chen, and Y.-T. Tsai, "Attention estimation system via smart glasses," in *2017 ieee conference on computational intelligence in bioinformatics and computational biology (cibcb)*. IEEE, 2017, pp. 1–5.

[22] P.-H. Tseng, I. G. Cameron, G. Pari, J. N. Reynolds, D. P. Munoz, and L. Itti, "High-throughput classification of clinical populations from natural viewing eye movements," *Journal of neurology*, vol. 260, pp. 275–284, 2013.

[23] N. M. Hanning, M. M. Himmelberg, and M. Carrasco, "Presaccadic attention depends on eye movement direction and is related to v1 cortical magnification," *bioRxiv*, 2023.

[24] M. Frutos-Pascual and B. Garcia-Zapirain, "Assessing visual attention using eye tracking sensors in intelligent cognitive therapies based on serious games," *Sensors*, vol. 15, no. 5, pp. 11 092–11 117, 2015.

[25] P. Deans, L. O'Laughlin, B. Brubaker, N. Gay, D. Krug *et al.*, "Use of eye movement tracking in the differential diagnosis of attention deficit hyperactivity disorder (adhd) and reading disability," *Psychology*, vol. 1, no. 04, p. 238, 2010.

[26] R. Krueger, S. Koch, and T. Ertl, "Saccadelenses: interactive exploratory filtering of eye tracking trajectories," in *2016 IEEE Second Workshop on Eye Tracking and Visualization (ETVIS)*. IEEE, 2016, pp. 31–34.

[27] H. C. Vazquez, "Artificial neuropsychology: Are large language models developing executive functions?" *arXiv preprint arXiv:2305.04134*, 2023.

[28] E. C. G. Merchán and M. Molina, "A machine consciousness architecture based on deep learning and gaussian processes," in *Hybrid Artificial Intelligent Systems: 15th International Conference, HAIS 2020, Gijón, Spain, November 11-13, 2020, Proceedings 15*. Springer, 2020, pp. 350–361.

[29] C.-M. Chen, J.-Y. Wang, and C.-M. Yu, "Assessing the attention levels of students by using a novel attention aware system based on brainwave signals," *British Journal of Educational Technology*, vol. 48, no. 2, pp. 348–369, 2017.

[30] C. K. Toa, K. S. Sim, and S. C. Tan, "Electroencephalogram-based attention level classification using convolution attention memory neural network," *IEEE Access*, vol. 9, pp. 58 870–58 881, 2021.

[31] A. Belle, R. H. Hargraves, K. Najarian *et al.*, "An automated optimal engagement and attention detection system using electrocardiogram," *Computational and mathematical methods in medicine*, 2012.

[32] M. Barz and D. Sonntag, "Automatic visual attention detection for mobile eye tracking using pre-trained computer vision models and human gaze," *Sensors*, vol. 21, no. 12, p. 4143, 2021.

[33] S. Gil, "El funcionamiento cognitivo en la vejez: atención y percepción en el adulto mayor," 2008.

[34] S. E. Petersen and M. I. Posner, "The attention system of the human brain: 20 years after," *Annual review of neuroscience*, vol. 35, no. 1, pp. 73–89, 2012.

[35] J. Driver and C. Spence, "Spatial synergies between auditory and visual attention," *Attention and performance XV: Conscious and nonconcious information processing*, 1994.

[36] R. Desimone, J. Duncan *et al.*, "Neural mechanisms of selective visual attention," *Annual review of neuroscience*, vol. 18, no. 1, pp. 193–222, 1995.

[37] J. T. Serences, *Voluntary and stimulus-driven attentional control in human cortex*. The Johns Hopkins University, 2006.

[38] L. Tamm, J. N. Epstein, J. L. Peugh, P. A. Nakonezny, and C. W. Hughes, "Preliminary data suggesting the efficacy of attention training for school-aged children with adhd," *Developmental cognitive neuroscience*, vol. 4, pp. 16–28, 2013.

[39] M. Varkanitsa, E. Godecke, and S. Kiran, "How much attention do we pay to attention deficits in poststroke aphasia?" *Stroke*, vol. 54, no. 1, pp. 55–66, 2023.

[40] J. Rönnberg, A. Sharma, C. Signoret, T. A. Campbell, and P. Sörqvist, "Cognitive hearing science: Investigating the relationship between selective attention and brain activity," *Frontiers in Neuroscience*, vol. 16, p. 1098340, 2022.

[41] A. T. Duchowski and A. T. Duchowski, *Eye tracking methodology: Theory and practice*. Springer, 2017.

[42] S. Ballesteros, "La atención selectiva modula el procesamiento de la información y la memoria implícita," *Acción psicológica*, vol. 11, no. 1, pp. 7–20, 2014.

[43] M. I. Posner, S. E. Petersen *et al.*, "The attention system of the human brain," *Annual review of neuroscience*, vol. 13, no. 1, pp. 25–42, 1990.

[44] S. P. Liversedge and J. M. Findlay, "Saccadic eye movements and cognition," *Trends in cognitive sciences*, vol. 4, no. 1, pp. 6–14, 2000.

[45] A. JothiPrabha, R. Bhargavi, and B. D. Rani, "Prediction of dyslexia severity levels from fixation and saccadic eye movement using machine learning," *Biomedical Signal Processing and Control*, vol. 79, p. 104094, 2023.

[46] M. A. A. Fernández, *Inteligencia artificial para programadores con prisa*. Universo de letras, 2022.

[47] Z. Niu, G. Zhong, and H. Yu, "A review on the attention mechanism of deep learning," *Neurocomputing*, vol. 452, pp. 48–62, 2021.

[48] K. P. Murphy, *Machine learning: a probabilistic perspective*. MIT press, 2012.

[49] Z.-H. Zhou, *Machine learning*. Springer Nature, 2021.

[50] T. Hastie, R. Tibshirani, J. H. Friedman, and J. H. Friedman, *The elements of statistical learning: data mining, inference, and prediction*. Springer, 2009, vol. 2.

[51] D. W. Hosmer Jr, S. Lemeshow, and R. X. Sturdivant, *Applied logistic regression*. John Wiley & Sons, 2013.

[52] L. Breiman, J. Friedman, R. A. Olshen, and C. J. Stone, *Classification and regression trees*. Routledge, 2017.

[53] M. A. Hearst, S. T. Dumais, E. Osuna, J. Platt, and B. Scholkopf, "Support vector machines," *IEEE Intelligent Systems and their applications*, vol. 13, no. 4, pp. 18–28, 1998.

[54] J. H. Friedman, "Greedy function approximation: a gradient boosting machine," *Annals of statistics*, pp. 1189–1232, 2001.

[55] P. Geurts, D. Ernst, and L. Wehenkel, "Extremely randomized trees," *Machine learning*, vol. 63, pp. 3–42, 2006.

[56] C. Peng and Q. Cheng, "Discriminative ridge machine: A classifier for high-dimensional data or imbalanced data," *IEEE transactions on neural networks and learning systems*, vol. 32, no. 6, pp. 2595–2609, 2020.

[57] H. William *et al.*, *Econometric analysis fifth edition*. New York University, 2003.

[58] C. M. Bishop, *Pattern Recognition and Machine Learning*. Springer, 2006.

[59] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[60] D. M. W. Powers, "Evaluation: From precision, recall and f-measure to roc, informedness, markedness and correlation," *Journal of Machine Learning Technologies*, vol. 2, no. 1, pp. 37–63, 2011. [Online]. Available: https://link.springer.com/article/10.1007/s10115-011-0391-2

[61] I. Markoulidakis, G. Kopsiaftis, I. Rallis, and I. Georgoulas, "Multi-class confusion matrix reduction method and its application on net promoter score classification problem," in *Proceedings of the 14th PErvasive technologies related to assistive environments conference*, 2021, pp. 412–419.

[62] J. Cohen, "A coefficient of agreement for nominal scales," *Educational and Psychological Measurement*, vol. 20, no. 1, pp. 37–46, 1960.

[63] B. W. Matthews, "Comparison of the predicted and observed secondary structure of t4 phage lysozyme," *Biochimica et Biophysica Acta (BBA) - Protein Structure*, vol. 405, no. 2, pp. 442–451, 1975.

[64] Y.-c. Wu and J.-w. Feng, "Development and application of artificial neural network," *Wireless Personal Communications*, vol. 102, pp. 1645–1656, 2018.

[65] S.-C. Wang and S.-C. Wang, "Artificial neural network," *Interdisciplinary computing in java programming*, pp. 81–100, 2003.

[66] R. Reed and R. J. MarksII, *Neural smithing: supervised learning in feedforward artificial neural networks*. Mit Press, 1999.

[67] S. Albawi, T. A. Mohammed, and S. Al-Zawi, "Understanding of a convolutional neural network," in *2017 international conference on engineering and technology (ICET)*. Ieee, 2017, pp. 1–6.

[68] K. O'Shea and R. Nash, "An introduction to convolutional neural networks," *arXiv preprint arXiv:1511.08458*, 2015.

[69] P. Kim and P. Kim, "Convolutional neural network," *MATLAB deep learning: with machine learning, neural networks and artificial intelligence*, pp. 121–147, 2017.

[70] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, vol. 25, 2012, pp. 1097–1105.

[71] C. Shorten and T. M. Khoshgoftaar, "A survey on image data augmentation for deep learning," *Journal of big data*, vol. 6, no. 1, pp. 1–48, 2019.

[72] B. S. Everitt, S. Landau, M. Leese, and D. Stahl, *Cluster Analysis*. Wiley, 2011.

[73] R. Gontijo-Lopes, S. J. Smullin, E. D. Cubuk, and E. Dyer, "Affinity and diversity: Quantifying mechanisms of data augmentation," *arXiv preprint arXiv:2002.08973*, 2020.

[74] L. v. d. Maaten and G. Hinton, "Visualizing data using t-sne," *Journal of machine learning research*, vol. 9, no. Nov, pp. 2579–2605, 2008.

[75] P. Liashchynskyi and P. Liashchynskyi, "Grid search, random search, genetic algorithm: a big comparison for nas," *arXiv preprint arXiv:1912.06059*, 2019.

[76] J. Bergstra and Y. Bengio, "Random search for hyper-parameter optimization," *The journal of machine learning research*, vol. 13, no. 1, pp. 281–305, 2012.

[77] B. Shahriari, K. Swersky, Z. Wang, R. P. Adams, and N. De Freitas, "Taking the human out of the loop: A review of bayesian optimization," *Proceedings of the IEEE*, vol. 104, no. 1, pp. 148–175, 2015.

[78] I. Syarif, A. Prugel-Bennett, and G. Wills, "Svm parameter optimization using grid search and genetic algorithm to improve classification performance," *TELKOMNIKA (Telecommunication Computing Electronics and Control)*, vol. 14, no. 4, pp. 1502–1509, 2016.

[79] Lenovo, "Lenovo legion y530 con intel core i7-8750h y 8gb de ram," 2018, computadora portátil.

[80] T. Kluyver, B. Ragan-Kelley, F. Pérez, B. E. Granger, M. Bussonnier, J. Frederic, K. Kelley, J. Hamrick, J. Grout, S. Corlay, P. Ivanov, D. Avila, S. Abdalla, and C. Willing, "Jupyter notebooks – a publishing format for reproducible computational workflows," *Positioning and Power in Academic Publishing: Players, Agents and Agendas*, pp. 87–90, 2016. [Online]. Available: https://doi.org/10.3233/978-1-61499-649-1-87

[81] E. Bisong, "Google colaboratory," in *Building Machine Learning and Deep Learning Models on Google Cloud Platform.* Apress, Berkeley, CA, 2019. [Online]. Available: https://doi.org/10.1007/978-1-4842-4470-8_7

[82] C. R. Harris, K. J. Millman, S. J. van der Walt, R. Gommers, P. Virtanen, D. Cournapeau, E. Wieser, J. Taylor, S. Berg, N. J. Smith, R. Kern, M. Picus, S. Hoyer, M. H. van Kerkwijk, M. Brett, A. Haldane, J. F. del Río, M. Wiebe, P. Peterson, P. Gérard-Marchant, K. Sheppard, T. Reddy, W. Weckesser, H. Abbasi, C. Gohlke, and T. E. Oliphant, "Array programming with NumPy," *Nature*, vol. 585, no. 7825, pp. 357–362, Sep. 2020. [Online]. Available: https://doi.org/10.1038/s41586-020-2649-2

[83] J. D. Hunter, "Matplotlib: A 2d graphics environment," *Computing in Science & Engineering*, vol. 9, no. 3, pp. 90–95, 2007.

[84] M. Ali, *PyCaret: An open source, low-code machine learning library in Python*, April 2020, pyCaret version 1.0.0. [Online]. Available: https://www.pycaret.org

[85] G. Van Rossum, *The Python Library Reference, release 3.8.2.* Python Software Foundation, 2020.

[86] M. L. Waskom, "seaborn: statistical data visualization," *Journal of Open Source Software*, vol. 6, no. 60, p. 3021, 2021. [Online]. Available: https://doi.org/10.21105/joss.03021

[87] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng, "TensorFlow: Large-scale machine learning on heterogeneous systems," 2015, software available from tensorflow.org. [Online]. Available: https://www.tensorflow.org/

[88] W. McKinney *et al.*, "Data structures for statistical computing in python," in *Proceedings of the 9th Python in Science Conference*, vol. 445. Austin, TX, 2010, pp. 51–56.

[89] M. del Carmen Cabrera-Hernández, C. A. García-Ezquerra, M. A. Aceves-Fernández, J. C. Pedraza-Ortega, and S. Tovar-Arriaga, "A dataset on eye movement tracking during the resolution of neuropsychological tests on a screen," *Data in Brief*, vol. 55, p. 110601, 2024.

[90] M. A. Aceves-Fernandez and M. d. C. Cabrera Hernandez, "Tdah repository," 2024, data set. [Online]. Available: https://doi.org/10.5281/zenodo.10519124

[91] P. Cheng, S. Grover, W. Wen, S. Sankaranarayanan, S. Davies, J. Fragetta, D. Soto, and R. M. Reinhart, "Dissociable rhythmic mechanisms enhance memory for conscious and nonconscious perceptual contents," *Proceedings of the National Academy of Sciences*, vol. 119, no. 44, p. e2211147119, 2022.

[92] R. Reinhart and P. X. Cheng, "Dissociable rhythmic mechanisms enhance memory for conscious and nonconscious perceptual contents," October 14 2022. [Online]. Available: https://doi.org/10.17605/OSF.IO/FQXK9

[93] A. Selivanova and P. F. Krabbe, "Eye tracking to explore attendance in health-state descriptions," *Plos one*, vol. 13, no. 1, p. e0190111, 2018.

[94] A. Selivanova, "Eye tracking in health-state descriptions," 2017, dataset. [Online]. Available: https://doi.org/10.6084/m9.figshare.5624974.v2

[95] L.-M. Vortmann, S. Ceh, and F. Putze, "Multimodal eeg and eye tracking feature fusion approaches for attention classification in hybrid bcis," *Frontiers in Computer Science*, vol. 4, p. 780580, 2022.

[96] S. M. Ceh, S. Walcher, C. Körner, C. Rominger, S. E. Kober, A. Fink, and M. Benedek, "Data for: Neurophysiological indicators of internal attention: an eeg-eye-tracking co-registration study," March 4 2024. [Online]. Available: https://doi.org/10.17605/OSF.IO/5U6R9

[97] Instituto Nacional del Derecho de Autor (INDAUTOR), "Sistema Inteligente de Pruebas para Detección de Niveles de Atención," Patent, 2022, registration number: 03-2022-071113515900-01.

[98] R. A. Cohen and R. A. Cohen, "Neuropsychological assessment of attention," *The neuropsychology of attention*, pp. 307–327, 1993.

[99] E. Anstey, "Development of the dominoes d48 test," Internal military publication, 1944, elaborated for the British Army during World War II for personnel selection purposes. No specific publication detail available.

[100] D. D. Drevon, R. M. Knight, and S. Bradley-Johnson, "Nonverbal and language-reduced measures of cognitive ability: A review and evaluation," *Contemporary School Psychology*, vol. 21, pp. 255–266, 2017.

[101] A. Campos, "Spatial imagery: A new measure of the visualization factor," *Imagination, cognition and personality*, vol. 29, no. 1, pp. 31–39, 2009.

[102] B. Csapó *et al.*, "Development of inductive reasoning in students across school grade levels," *Thinking Skills and Creativity*, vol. 37, p. 100699, 2020.

[103] C. M. Musil, C. B. Warner, P. K. Yobas, and S. L. Jones, "A comparison of imputation techniques for handling missing data," *Western journal of nursing research*, vol. 24, no. 7, pp. 815–829, 2002.

[104] S. McKinley and M. Levine, "Cubic spline interpolation," *College of the Redwoods*, vol. 45, no. 1, pp. 1049–1060, 1998.

[105] C. De Boor, *A Practical Guide to Splines*. New York: Springer-Verlag, 1978.

[106] P. H. C. Eilers and B. D. Marx, "Flexible smoothing with b-splines and penalties," *Statistical Science*, vol. 11, no. 2, pp. 89–121, 1996.

[107] L. L. Schumaker, *Spline Functions: Basic Theory*, 3rd ed. Cambridge: Cambridge University Press, 2007.

[108] P. Dierckx, "Curve and surface fitting with splines," *Monographs on Numerical Analysis*, 1993.

[109] S. C. Chapra, R. P. Canale, R. S. G. Ruiz, V. H. I. Mercado, E. M. Díaz, and G. E. Benites, *Métodos numéricos para ingenieros*. McGraw-Hill New York, NY, USA, 2011, vol. 5.

[110] D. D. Salvucci and J. H. Goldberg, "Identifying fixations and saccades in eye-tracking protocols," in *Proceedings of the 2000 symposium on Eye tracking research & applications*, 2000, pp. 71–78.

[111] T. Blascheck, K. Kurzhals, M. Raschke, M. Burch, D. Weiskopf, and T. Ertl, "State-of-the-art of visualization for eye tracking data." in *Eurovis (stars)*, 2014, p. 29.

[112] R. Matos, "Designing eye tracking experiments to measure human behavior," *Merriënboer, JJG van, and Sweller, J.(2005). Cognitive load theory and complex learning: Recent developments and future directions. Educational Psychology Review*, vol. 17, 2010.

[113] A. Poole and L. J. Ball, "Eye tracking in human-computer interaction and usability research: Current status and future prospects," in *Encyclopedia of human-computer interaction*. Idea Group Reference, 2005, pp. 211–219.

[114] R. J. Jacob and K. S. Karn, "Eye tracking in human-computer interaction and usability research: Ready to deliver the promises," in *The mind's eye*. Elsevier, 2003, pp. 573–605.

[115] M. A. Just and P. A. Carpenter, "Eye fixations and cognitive processes," *Cognitive psychology*, vol. 8, no. 4, pp. 441–480, 1976.

[116] K. Rayner, "Eye movements in reading and information processing: 20 years of research," *Psychological bulletin*, vol. 124, no. 3, p. 372, 1998.

[117] B. N. Türkan, S. Amado, E. S. Ercan, and I. Perçinel, "Comparison of change detection performance and visual search patterns among children with/without adhd: Evidence from eye movements," *Research in developmental disabilities*, vol. 49, pp. 205–215, 2016.

[118] Z.-F. Shi, C. Zhou, W.-L. Zheng, and B.-L. Lu, "Attention evaluation with eye tracking glasses for eeg-based emotion recognition," in *2017 8th International IEEE/EMBS Conference on Neural Engineering (NER)*. IEEE, 2017, pp. 86–89.

[119] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Going deeper with convolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 1–9.

[120] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[121] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 1251–1258.

[122] G. Bradski, "The opencv library," 2000, available at https://opencv.org/. [Online]. Available: https://opencv.org/

[123] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*. MIT Press, 2016.

[124] Y. LeCun, L. Bottou, G. B. Orr, and K.-R. Müller, "Efficient backprop," in *Neural networks: Tricks of the trade*. Springer, 2012, pp. 9–48.

[125] M. Sonka, V. Hlavac, and R. Boyle, *Image processing, analysis, and machine vision*. Cengage Learning, 2008.

[126] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[127] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[128] P. Santana Vidal, N. Cid Rivera, C. Pinilla Navarro, and S. Quezada Gutierrez, "Atención selectiva, atención sostenida, inhibición y flexibilidad cognitiva en niñas y adolescentes de 12 a 14 años con tdah predominio de falta de atención," Ph.D. dissertation, Universidad Católica de la Santísima Concepción, 2016.

[129] C. D. Wickens, J. S. McCarley, and R. S. Gutzwiller, *Applied attention theory*. CRC press, 2022.

Annexes