Augmentative and Alternative Communication Using Eye Tracking and Word Recommendation Using Language Models

Bruno Waideman, and Plinio Thomaz Aquino Junior

***Abstract*—The production, storage, and dissemination of information have evolved from ancient communication methods to modern digital technologies, with digital media playing a key role in connecting individuals. While keyboards are common tools for interaction, they present challenges for individuals with motor impairments. Augmentative and Alternative Communication (AAC) techniques, including gesture input, voice commands, and sensor-based systems, have emerged to address these limitations. Eye tracking, used in accessibility systems, offers both opportunities and challenges, such as visual fatigue and inaccuracies that lead to slower typing. To address these challenges, this study proposes an interaction approach integrating eye movement tracking with a virtual keyboard, utilizing an artificial neural network to interpret gaze data and translate intentions within the interface at a low cost for the user. Additionally, a Language Model (LM) aids in predicting next-word suggestions. This research will assess the impact of these technologies on typing speed, error rate, and linguistic predictability, contributing both scientifically and societally to the advancement of accessible communication systems.**

Link to graphical and video abstracts, and to code: https://latamt.ieeer9.org/index.php/transactions/article/view/9596

***Index Terms*—AAC, Eye tracking, Virtual keyboard, Artificial neural network, LM, Accessibility.**

I. Introduction

INTERACTIONS with computers traditionally rely on keyboards and mice, which require hand use [1]. This can create barriers for individuals with motor impairments. Alternatively, new interaction methods, such as gesture input, voice commands, and sensor-based inputs, are becoming increasingly common [2].

Assistive Technology (AT) aims to address functional challenges for individuals with disabilities, enhancing mobility, self-care, social inclusion, and reducing activity limitations [3]. A subset of AT, AAC, improves communication by incorporating gestures, sounds, and facial and body expressions to convey needs and opinions [4]. AAC is particularly effective for individuals with unintelligible or limited speech [5].

AAC integrates four strategies: symbols, resources, techniques, and strategies for enabling communication [6]. Symbols include gestures, images, or sounds that represent words or messages. Resources involve devices such as mobile phones and tablets [7]. Communication boards, where symbols are used to construct messages, are commonly used due to their affordability [8]. Leveraging commonly available hardware, such as webcams, can make AAC systems more accessible [9]. Techniques and strategies focus on improving the speed and accuracy of communication.

Eye-tracking technology is one such tool used in AAC systems, helping reduce visual fatigue and improve typing speed [10]. However, eye-tracking presents challenges such as high equipment cost and accuracy limitations [2]. Advances in Artificial Intelligence (AI), have improved the accessibility of these systems by utilizing common devices for Human-Computer Interaction (HCI) [11]. Classifier algorithms, including neural networks, are increasingly enhancing AAC systems' performance.

Auxiliary-assisted interfaces in AAC improve user interaction by increasing speed and offering hygienic input methods [12]. Key factors include key layout, size, distribution, and feedback [5], while error prediction and correction are especially important for eye-typing. Although AAC promotes inclusive HCI, digital communication remains inaccessible to many with motor impairments. The integration of eye-tracking and AI offers opportunities to enhance accuracy, reduce fatigue, and improve usability, contributing to greater autonomy and accessibility.

This work aims to develop an AAC system using eye-tracking and head orientation to control a virtual keyboard, targeting users with severe motor disabilities. The system allows users to select letters and words using eye and head movements, providing an intuitive and low-cost (without the use of sensors) system for written communication.

The secondary goal is to integrate a LM into the AAC system, offering real-time word suggestions based on context. The study evaluates the impact of the LM on system efficiency, measuring typing speed, communication accuracy, and user satisfaction.

The paper is structured as follows: Section I introduces the research objectives. Section II reviews relevant literature and evaluation metrics. Section III describes the methodology and experimental design. Section IV presents the results, and Section V summarizes findings and suggests future research directions.

The associate editor coordinating the review of this manuscript and approving it for publication was Samuel Ortega (*Corresponding author: Bruno Waideman*).

This work was financially supported by the Centro Universitário FEI.

Bruno Waideman, and P. T. Aquino Jr. are with Centro Universitário FEI, São Bernardo do Campo, Brazil (e-mails: bruno.waideman@gmail.com, and plinio@fei.edu.br).

II. Literature Review

The literature review examines studies on eye-controlled interaction, focusing on virtual keyboards for users with motor impairments. It explores technologies and methodologies to improve HCI efficiency and accessibility.

Hori, Sakano, and Saitoh (2004), along with Wobbrock et al. (2007) and Billing, Roggen, and Tröster (2008), introduced key techniques in eye tracking to enhance cursor control and selection on virtual keyboards, demonstrating advancements in gesture recognition and eye-guided cursor movement [13]–[15].

Continuing this trajectory, Bee et al. (2008) and Majaranta, Ahola, and Špakov (2009) investigate the usability and efficiency of eye-controlled writing interfaces, employing adaptations of existing typing methods to suit eye movements [16], [17]. Their findings underscore the importance of customizing interaction techniques to leverage the unique capabilities of eye-tracking technology.

Further innovations are presented by Tangsuksant et al. (2012), Saraswati, Sigit, and Harsono (2016), and Cecotti, Meena, and Prasad (2018), who explore advanced eye-tracking systems and gesture recognition algorithms to refine virtual keyboard interfaces [18]–[20]. These studies collectively emphasize the need for robust, real-time processing capabilities to translate eye movements into digital inputs accurately.

Recent contributions by Tantisatirapong and Phothisonothai (2018) and Rusydi et al. (2019) focus on optimizing the layout and responsiveness of virtual keyboards to accommodate the specific needs of users [21], [22]. These enhancements are crucial for reducing the cognitive and physical strain on users, further enabling their participation in digital communications.

Attiah and Khairullah (2021) and Silva and Paschoarelli Veiga (2021) propose the integration of multimodal inputs and advanced language processing algorithms to refine eye-controlled virtual keyboards [2], [23]. Their research highlights the ongoing need for interface improvements and the application of artificial intelligence to interpret user intentions better and facilitate more natural and efficient communication methods.

Anandika et al. (2023) constructed a virtual keyboard combined with a leap motion sensor and a neural network, achieving 99% and an average time of 5.45 seconds per character [24]. This approach demonstrates the versatility of possible inputs to construct a virtual keyboard.

Lastly, Shaima et al. (2024) analyze Neuralink and explore its methodology, which involves implanting ultra-thin electrodes in the brain to capture and transmit neural signals [25]. However, the authors also warn of ethical and security challenges.

This body of literature underscores the technological advancements in AAC. It highlights the continuous need for research combining hardware capabilities with sophisticated software algorithms to create more intuitive and accessible communication aids.

Research on AAC solutions for motor-impaired individuals encompasses analog, electrode-based, and video-based systems. While contact-based methods offer high precision and low latency, they are less common due to invasiveness and cost [26]. In contrast, image and video-based approaches, though requiring complex real-time processing, have improved in accuracy and efficiency with technological advancements [2].

Recent developments integrate video with Convolutional Neural Network (CNN) and probabilistic models to enhance accuracy and typing speed. The trend emphasizes artificial intelligence and optimized processing to improve assistive keyboards. The primary goals are to translate subtle eye movements into user intentions and enhance typing productivity through alternative input methods.

Thus, the aim is to use video, combined with a CNN algorithm as seen in the work of [23] for information processing. For the recommendation of letters or words, an N-gram algorithm will be used, similar to the one adopted in [2]. As for metrics, the time per character, the success rate (to calculate errors), and the evaluation of tests with volunteers will be utilized.

III. AAC Using Eye Tracking and Word Recommendation

This work proposes a user interaction system using a virtual keyboard interface controlled by gaze tracking via Eye Tracking, processed by a CNN. The system also supports word insertion and recommendation, powered by a corpus and LM. Fig. 1 shows the architecture, divided into seven stages. The data collection and prediciton were performed on a computer with an Intel i5, 16 GB of RAM, and an Nvidia GeForce RTX 3050. Training processes used a Linux Gold 5118 with 188 GB of RAM, two Nvidia Tesla V100 GPUs and CUDA 11.8.

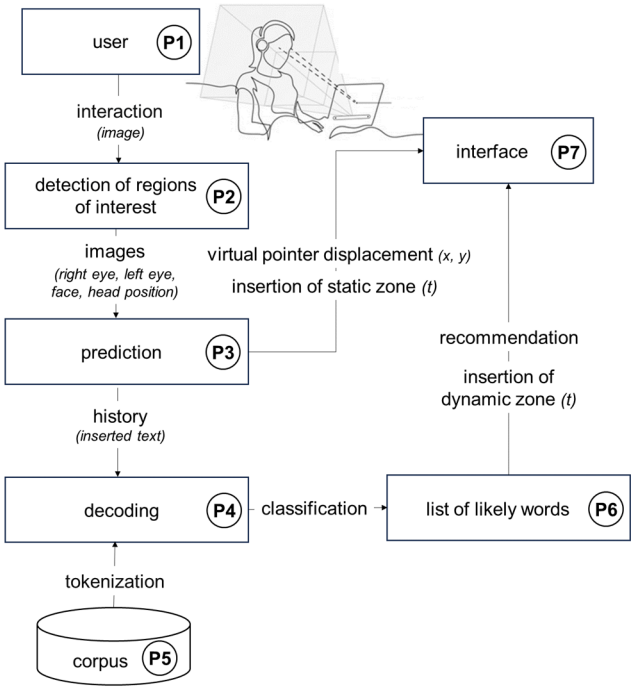

Fig. 1. System architecture for AAC, utilizing computer vision, CNNs, and LMs.

A. User

In Stage P1, user interaction occurs in a controlled hospital setting to enhance system performance. Patients use AAC devices to communicate when speech or typing is impaired [5]. Although digital AAC systems are advancing, they demand accurate eye and head control. The system design accounts for involuntary movements and minimizes training data requirements. Optimal use involves the patient seated 40 cm from a vertical monitor with a webcam aligned above the screen [27]. No additional sensors are needed. The keyboard supports sentence construction via alphabetic input, enabling communication of needs and daily interactions.

B. Detection of Regions of Interest

In stage P2, images are captured in real-time and stored in a queue to prevent processing delays. The system must then isolate the face from other image elements, applying the MMOD-CNN deep learning method with the stored image as input [28]. The objective is to achieve higher scores for correct labeling compared to incorrect ones. The cost function $\mathcal{L}$ is defined as follows, where C is the scaling factor, n the total number of examples, $\triangle$ the loss, F the prediction function, and (x_i, y_i) a randomly selected image pair [28]:

$$\mathcal{L} = \frac{C}{n} \sum_{i=1}^{n} \triangle(\arg\max_{y \in Y} F(x_i, y), y_i) \tag{1}$$

This equation calculates the weighted average of classification losses across training examples to minimize errors and detect facial regions of interest. From facial landmarks, features are extracted using a five-point model to identify the eyes and nose. The "right eye" and "left eye" classes are used to determine head angle, while the "head position" class is represented by a black rectangle on a blank image. All images have a resolution of 65×65 to optimize dataset size and model performance.

Involuntary eye movements, such as blinking or attention shifts, can affect accuracy. Studies report that blinking may lead to a 2% data loss due to occlusion of key regions, compromising eye-tracking performance [27], [29]. To reduce this impact, a sufficiently large sample must be collected during training.

C. Calibration and Data Collection

Following region-of-interest detection, the system requires labeled data obtained through calibration and data collection modules. The calibration module captures (x, y) coordinates and class labels by prompting the user to focus on nine predefined screen points, confirming each with a keyboard input for accurate self-labeling. Subsequently, the data collection module presents a continuously moving target, recording gaze and head orientation data in milliseconds as the user tracks it. This process gathers essential information for model training.

D. Prediction

After collecting the labels, a CNN is trained to predict the x and y coordinates on the interface. The proposed CNN architecture, illustrated in Fig. 2, is structured into four distinct sub-networks, each processing different types of input data.

All sub-networks receive input images of size 64×64. The first three sub-networks utilize RGB channels to represent the face, right eye, and left eye, respectively, while the fourth sub-network uses a single channel to represent head position. In the subsequent stage, convolution operations are applied, as defined in Equation 2:

$$(f * g)(k) = h(k) = \sum_{j=0}^{k} f(j) \cdot g(k - j) \tag{2}$$

f and g are sequences, and k provides the index of the output. The equation describes the convolution process for the sequences, with dimensions depending on the input sizes.

Sub-networks 1, 2, and 3 pass through a second convolutional layer, followed by the ReLU activation function, which mitigates issues such as vanishing gradients, as described in Equation 3.

$$f(x) = max(0, x) \tag{3}$$

Batch normalization is then applied to standardize layer outputs, calculated by Equations 4 and 5:

$$\mu = \frac{1}{m} \sum_{i} H_i \tag{4}$$

$$\sigma = \sqrt{\delta + \frac{1}{m} \sum_{i} (H - \mu)_i^2} \tag{5}$$

H represents the mini-batch activations, μ the mean, and σ the standard deviation for normalization, ensuring stable gradient propagation [30].

Next, max pooling is applied to provide translation invariance, followed by additional convolution layers for sub-network 1. The sub-networks are then merged and flattened into 64 fully connected layers of 128 nodes. Linear functions and dropouts are used to prevent overfitting, described by Equation 6:

$$r_j^{(l)} \sim Bernoulli(p), \tag{6}$$

$$\tilde{y}^{(l)} = r^{(l)} * y^{(l)}, \tag{7}$$

$$z_i^{(l+1)} = w_i^{(l+1)} \tilde{y}^l + b_i^{(l+1)}, \tag{8}$$

$$y_i^{(l+1)} = f(z_i^{(l+1)}). \tag{9}$$

where r is a vector of independent Bernoulli random variables, l is the index of the hidden layers of the neural network, $\tilde{y}$ is a vector of reduced outputs, y is the output vector of layer l, z is the input vector of layer l, w is the weight of layer l, b is the bias of layer l and f is the activation function.

At the end of the process, the x and y coordinates are predicted, representing the user's gaze position on the screen. After training, the CNN model is used with runtime inputs to enable user interaction. Depending on the predicted action, the pointer can either remain for a set time to trigger an insertion or move across the interface. The interaction and feedback are presented to the user in the final interface stage (P7).

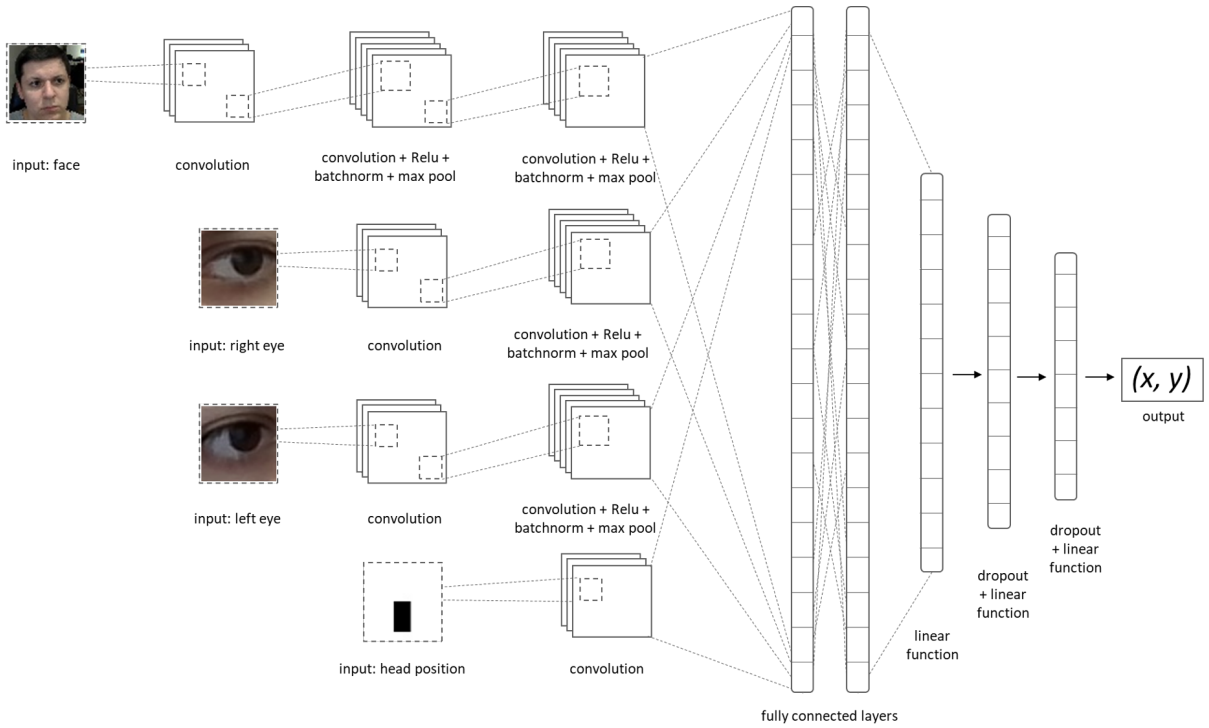

Fig. 2. CNN architecture generates (x, y) eye movement coordinates from face images using operations.

E. Decoding and List of Likely Words

In stage P4, decoding is performed using two input datasets: the last three words entered in the presentation zone and a corpus represented in stage P5. The corpus, was constructed from Folha de São Paulo news articles from 1994 and 1995, contains 42,109,286 tokens and 234,163 unique words, processed through simple tokenization. Based on these datasets, a trigram estimates the probability of the next word using relative word frequencies, as shown in Equation 10.

$$P_e(w_1|w_{i-2,i-1}) = \frac{count(w_{i-2,i})}{count(w_{i-2,i-1})} \tag{10}$$

P_e is the probability of the word w_1 occurring after the sequence $w_{i-2,i-1}$, and $count$ represents the occurrence count of the word sequence. A challenge with Equation 10 is that missing trigrams in the corpus result in a zero probability for the chain. To mitigate this, interpolation is applied, combining bigram and unigram probabilities, which, although less accurate than trigrams, help provide non-zero estimates.

After probability calculations, fitness-proportional selection (roulette selection) is used to choose words for the interface, as shown in Equation 11:

$$p_i = \frac{f_i}{\sum_{j=1}^{N} f'_j} \tag{11}$$

p_i represents the selection probability of word i, f_i is the calculated probability of word i, N is the total number of words, and f'_j is the word frequency. Based on this method, 10 possible next words are selected for recommendation, as described in P6.

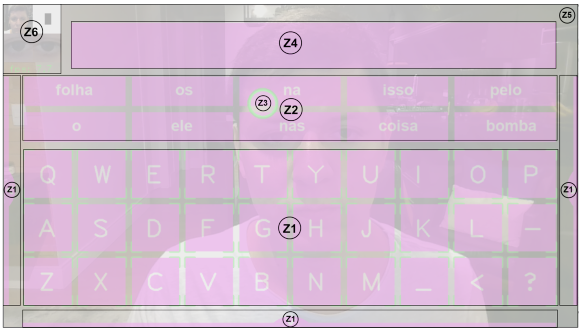

Fig. 3. User interacting with the assisted virtual keyboard, organized into six interaction zones.

F. Interface

In stage P7, the interface is organized and presented to the user, as shown in Fig. 3.

The first component is the static zone (Z1), which remains constant during interaction. It is divided into four regions. The central region features a QWERTY keyboard layout and the other three regions, contain functions for deleting all text, inserting a space, and deleting the last character. This approach is inspired by the "infinite edge" technique, originally seen in early graphical systems like the Commodore Amiga [31].

The second component is the dynamic recommendation zone (Z2), located above the static zone. It displays likely next words based on previous input, generated by a LM [32]. This zone enhances productivity by minimizing letter-by-letter input. Both the static and dynamic zones are navigated using a virtual pointer (Z3) controlled by the user's gaze and head movement.

The third component is the presentation zone (Z4), at the top of the interface, which displays the entered text and acts as a resting zone without allowing character input. Once the pointer reaches the target, the system inserts the character after a dwell time.

Additional feedback is continuously provided to support user interaction. A real-time webcam feed (Z5) and detected regions of interest (Z6), including the eyes and head, are displayed alongside frame rate data. The pointer size adapts based on prediction accuracy, enlarging when accuracy is low. Keys change color briefly to indicate interaction, while the pointer turns red during character insertion, then reverts to green, offering clear visual confirmation.

The system architecture and interface integrate eye tracking and head movement for seamless virtual keyboard interaction. By combining static and dynamic zones with continuous feedback, it enhances user experience and communication efficiency for individuals with motor impairments. These features establish a foundation for advancing assistive technologies, enabling more intuitive and adaptative human-computer interactions.

IV. Results

The experiments evaluated four main aspects: facial recognition accuracy, CNN precision measured by pixel error, recommendation system effectiveness based on the LM, and the usability of the integrated system by volunteers, focusing on prior knowledge, the dynamic recommendation zone, and dwell time.

A. Facial Recognition

The first experiment employed the Eye Chimera database, comprising 1,135 labeled images from 40 individuals [33]. Volunteers followed predefined movement patterns, and significant frames were extracted from video in 640×480 or 1920×1080 resolution. Images were processed and classified using algorithms, as shown in Fig. 4. The CNN-based model achieved 99.9% accuracy, correctly identifying 1,134 images. The only misclassification involved partial facial obstruction from eyeglasses, though similar images were correctly labeled. The model demonstrated robustness to variations in appearance, including hair, weight, skin tone, and eye shape

The second test structure assessed the model's prediction performance using a custom database generated from calibration and data collection procedures across interface points. Built from a single user's data, the dataset covered 99.8% of interface points and included labels for facial alignment, eye, and head positions. Importantly, no data from the Eye Chimera database was used, ensuring evaluation of model generalization. Data distribution is illustrated in Fig. 5, and the model architecture is shown in Fig. 2.

Two simplified class subsets were used for analysis. In the first, directional classes were grouped into *Right* and *Left*, yielding 93% overall accuracy (97% for *Right*, 87% for *Left*). In the second, groupings were made into *Up* and *Down*, achieving 76% accuracy (79% for *Up*, 73% for *Down*). Detailed metrics are provided in Table I.

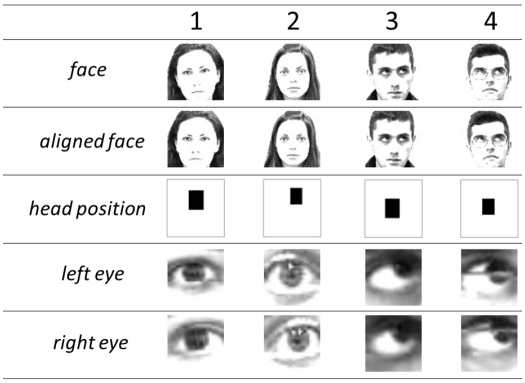

Fig. 4. Representation of the data classified by the facial recognition system.

TABLE I
Prediction Using the Eye Chimera Database as Input

class	samples	accuracy	errors	% accuracy
Right	406	393	13	97%
Left	404	353	51	87%
Up	278	221	57	79%
Down	259	188	71	73%

B. Neural Network

The second experiment utilized the same dataset described in the Facial Recognition subsection to evaluate four different models. The first model, referred to as the Misaligned Face Model, used images of the face without alignment preprocessing. The second, the Aligned Face Model, employed aligned facial images to improve consistency. The third model focused exclusively on eye-region data, using images of both eyes. Finally, the Multiple Inputs Model integrated various features, including misaligned face images, individual images of the right and left eyes, head position, and head angle, to enhance prediction accuracy.

All models are variations of those presented in Fig. 2. Models 1) and 2) use only the first subnetwork, differing only in the input image. This structure is followed by a linear function and a dropout layer. Model 3) integrates the second and third subnetworks, followed by a fully connected layer, a linear function and a dropout layer. Model 4) is the exact model presented.

The models were optimized using a systematic hyperparameter tuning process. An exploratory phase tested a wide range of hyperparameters, followed by fine-tuning using the ASHA algorithm [34]. The hyperparameters included batch size, learning rate, filter size, number of layers, and fully connected nodes. The performance curves of low-performing models converged quickly, validating the effectiveness of early elimination in the ASHA algorithm (see Fig. 6).

After training, models were evaluated using the Root Mean Square Error (RMSE) metric [35]:

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \tag{12}$$

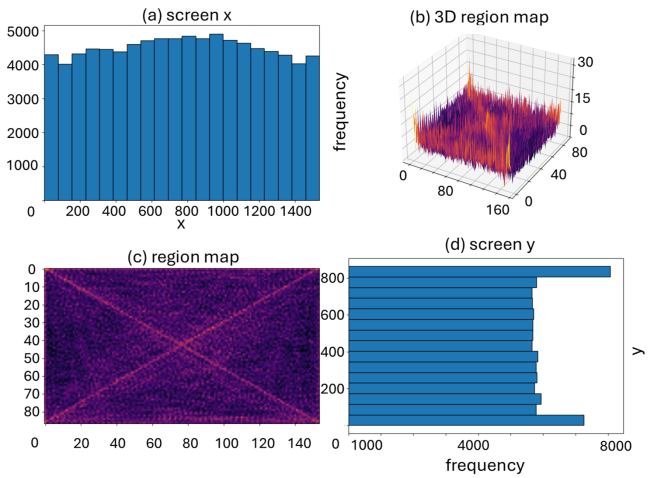

Fig. 5. Distribution of the proposed CNN training data. (a) Histogram of x-coordinates showing a uniform distribution. (b) Three-dimensional map indicating spatial variability. (c) Heatmap highlighting point concentration and interaction patterns. (d) Histogram of y-coordinates revealing variations in user interaction along the vertical axis.

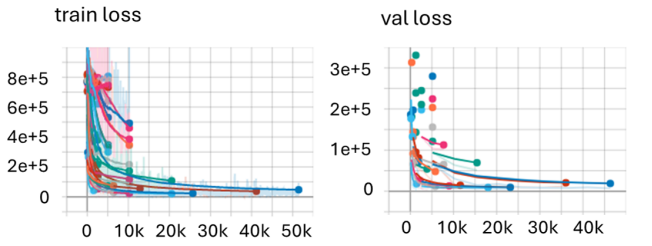

Fig. 6. Training and validation loss curves during hyperparameter tuning, illustrating the ASHA algorithm's efficient elimination of low-performing models.

The best model is the one with the lowest RMSE, indicating fewer prediction errors. The misaligned face model showed an RMSE of 44.60 pixels, while the aligned face model had 44.92 pixels, suggesting that head angle is important for tracking.

The eyes model, which used images from the right and left eye, performed better, with an RMSE of 41.75 pixels. This highlights the relevance of iris and pupil movements in tracking applications.

The multiple input model, which combined the misaligned face, both eyes, and head position, achieved the lowest RMSE of 31.58 pixels.

Fig. 7 illustrates the error distribution on the screen, measured in pixels and represented by a color scale on the right. The horizontal axis (screen x) indicates the error position along the screen's width, while the vertical axis (screen y) represents its height. Darker regions correspond to lower errors, indicating areas where the model exhibits better generalization capability and, consequently, higher prediction accuracy. In contrast, lighter regions indicate higher errors, suggesting that the model's predictions tend to be less precise in these areas. The overall distribution suggests effective generalization across the screen, with errors relatively evenly spread and only a few isolated high-intensity areas. These localized error

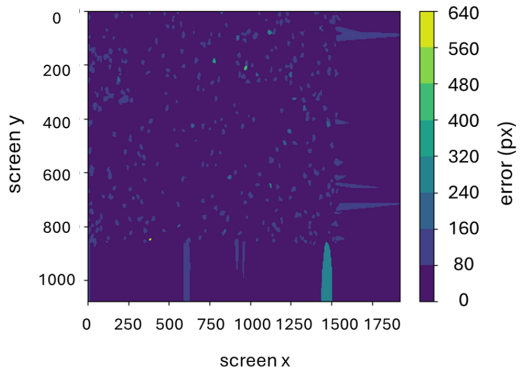

Fig. 7. Error distribution map on the screen, measured in pixels and represented by a color scale. Dark tones indicate lower errors, while light tones highlight regions with greater inaccuracy.

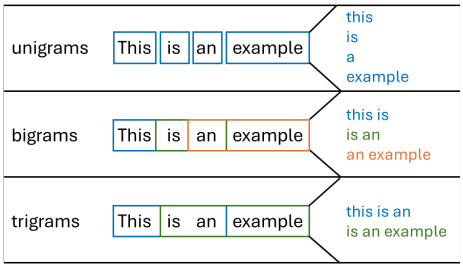

Fig. 8. Representation of n-grams for the sentence. Unigrams consider individual words, bigrams group consecutive pairs of words, and trigrams form sequences of three words.

concentrations may be associated with variations in user position, lighting conditions, or specific interface regions.

C. Language Model

In the third experiment, the LM was evaluated based on its ability to assign high probabilities to valid and syntactically correct sentences and low probabilities to incorrect or rare ones. The corpus Folha de Sao Paulo was divided into training sets (89%) and testing sets (11%). A simple separator was used to extract 1,000 non-repeating sentences for training, with lexical sizes ranging from 25 to 370 characters. To conduct the evaluation, probabilistic model classes were used, which assume that the probability of a future unit can be approximated by certain elements from the past. In this context, unigrams, bigrams and, finally, n-grams are employed, as shown in Fig. 8.

In mathematical terms, this logic can be represented by the following equation:

$$P(w_n|w_{n-N+1}^{n-1}) = \frac{C(w_{n-N+1}^{n-1}w_n)}{C(w_{n-N+1}^{n-1})} \tag{13}$$

The equation represents the conditional probability of a word w_n given a sequence of preceding words w_{n-N+1}^{n-1}. Let $C(w_{n-N+1}^{n-1}w_n)$ be the count of occurrences of the complete sequence $w_{n-N+1}^{n-1}w_n$ and $C(w_{n-N+1}^{n-1})$ be the count of occurrences of the subsequence w_{n-N+1}^{n-1}.

Table II shows the evolution of perplexity as the number of training tokens increased. The Δ values represent percentage

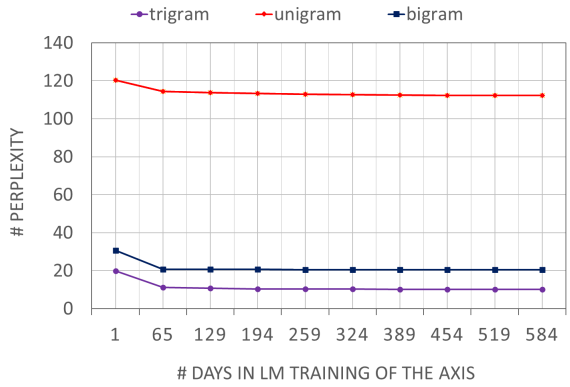

Fig. 9. Comparative evolution of the perplexity.

variations from the previous results. Perplexity is this case, can be described as the inverse probability of the test set, normalized by the number of words. For a test set $W = \{w_1 w_2 ... w_N\}$, the perplexity can be described as:

$$PP(W) = \sqrt[N]{\prod_{i=1}^{N} \frac{1}{P(w_i|w_1 ... w_{i-1})}} \tag{14}$$

$PP(W)$ is the perplexity of the word sequence W, N is the length of the sequence, w_i is the i-th word in the sequence, and $P(w_i|w_1 ... w_{i-1})$ is the conditional probability of the word w_i given the context.

As the number of training tokens increased, perplexity decreased, indicating improved model accuracy. The trigram model showed the best results, with a 53.7% improvement from the first to the last simulation. The bigram model achieved a 33.6% improvement, and the unigram model showed a 6.9% improvement. The trigram model reached a perplexity of 10.0, with a percentage difference of 91.1% compared to the unigram model.

Fig. 9 illustrates the perplexity evolution, where all models followed similar trends, though the unigram's results remained consistently higher.

Overall, the trigram model exhibited the best performance across the dataset, with substantial improvements as the training data increased.

D. Tests with Volunteers

The fourth experiment aimed to evaluate the system through tests with volunteers. The study was approved by the ethics committee under protocol CAAE 77015023.0.0000.5510 on March 7, 2024. All volunteers were informed of the research objectives, methods, and data collection process. Participation was contingent on signing the Informed Consent Form (ICF).

Five volunteers aged 18 to 30, without motor difficulties, were selected. Each volunteer participated in two 90-minute sessions, during which no personal devices were used. In the first session, volunteers completed a Pre-Questionnaire to verify eligibility and identify personal characteristics (e.g., height, typing method) that could influence system use. Exclusion criteria included previous experience with eye trackers or using facial-obstructing items.

Volunteers were seated 40 cm from the monitor, and their interactions were recorded. Calibration and data collection involved 10-minute battery tasks, with breaks allowed.

In the second session, volunteers completed four test batteries (denoted as Seq), typing four target phrases in each, resulting in a total of 16 phrases (denoted as f), 8 of which were unique. The character count of the phrases ranged from 24 to 49, with a total of 522 characters typed per volunteer, or 2,610 characters when considering all volunteers. Table III shows the average results.

To comprehend the result, two concepts are essential. The precision rate is calculated by summing the incorrectly entered letters and words (when compared to the target letter or word) and dividing this sum by the total number of characters in the target sentence. The success rate, on the other hand, is determined by comparing the number of correctly typed characters to the number of incorrectly typed characters in the final sentence.

The results obtained in this study indicate an average time of 7.3 seconds per character, representing a significant improvement compared to previous studies. Saraswati, Sigit, and Harsono (2016) reported an average time of 14.8 seconds per character and employed the Haar Cascade technique combined with the integral projection method for decision-making. Interaction in their system was strictly performed through vertical and horizontal head movements, and the virtual keyboard was simplified by assigning three letters to a single key. Rusydi et al. (2019) reported an average of 18 seconds per character, using EOG device integrated with an adaptive keyboard. In contrast, the solution proposed in this study adopted others technologies and techniques, such as CNNs applied to facial recognition and user intent prediction, as well as volunteer-specific training. Additionally, a virtual keyboard with a QWERTY layout was used, along with the implementation of a word recommendation system based on a language model, which may have contributed to the significant reduction in character input time. In contrast, Cecotti, Meena, and Prasad (2018) obtained a result of 6.42 seconds (SD 0.1) and Anandika et al. (2023) achieved a result of 5.5 seconds. Despite the differences in time, it is important to note that both studies utilized sensors to support eye tracking, whereas this study aimed to develop a low-cost device, relying solely on a computer and a webcam. This results suggest that the use of more accessible technologies may represent a viable alternative.

Additionally, a precision rate of 93% (SD 6%), and a success rate of 100% (SD 0%) were recorded. Despite errors averaging 7% per phrase, all volunteers completed the target phrases.

Three hypothesis tests were conducted using Time per Character and Precision Rate metrics:

1. Prior Knowledge/Practice: Comparing results between batteries 1-2 and 3-4, time per character improved by 21%, and precision rate by 4.8%, indicating that prior knowledge or practice benefited performance.

2. Dynamic Recommendation Zone (DRZ): When the DRZ was enabled (batteries 2-3), time per character improved by

TABLE II
MEASUREMENT OF THE AVERAGE PERPLEXITY OF MODELS TRAINED WITH DIFFERENT NUMBERS OF TOKENS

Days	Tokens	Distinct Words	1gram	Δ 1gram	2gram	Δ 2gram	3gram	Δ 3gram	3gram vs 1gram
1	56,478	8,600	120.3	-	30.7	-	19.8	-	-83.6%
65	4,969,620	94,205	114.3	-5.0%	20.7	-32.5%	11.1	-44.1%	-90.3%
129	9,489,539	127,142	113.8	-0.4%	20.7	0.0%	10.7	-3.7%	-90.6%
194	13,407,963	149,693	113.3	-0.5%	20.6	-0.7%	10.4	-2.5%	-90.8%
259	17,163,336	167,349	112.9	-0.4%	20.5	-0.2%	10.4	-0.2%	-90.8%
324	21,016,255	182,634	112.7	-0.2%	20.5	-0.1%	10.3	-0.9%	-90.9%
389	24,692,275	195,833	112.8	-0.1%	20.5	0.0%	10.1	-1.4%	-91.0%
454	29,105,286	210,239	112.3	-0.2%	20.5	-0.2%	10.1	0.0%	-91.0%
519	33,389,403	222,744	112.2	-0.1%	20.5	0.0%	10.0	-0.8%	-91.1%
584	37,655,257	234,163	112.2	-0.1%	20.5	0.0%	10.0	0.0%	-91.1%

TABLE III
ANALYSIS OF THE TESTS WITH VOLUNTEERS

Seq	DRZ	tp	f	Time per Character(s)	Precision Rate	Success Rate
1	Disabled	t1	f1	8.2	87.9%	100%
			f2	7.3	91.0%	100%
			f3	8.1	76.3%	100%
			f4	7.6	85.0%	100%
2	Enabled	t2	f5	7.9	97.7%	100%
			f6	8.6	100.0%	100%
			f7	6.8	98.5%	100%
			f8	8.8	92.7%	100%
3	Enabled	t1	f3	7.7	88.9%	100%
			f4	5.7	95.0%	100%
			f7	4.1	99.2%	100%
			f8	6.1	91.4%	100%
4	Disabled	t2	f1	7.2	99.3%	100%
			f2	6.6	97.2%	100%
			f5	8.4	98.3%	100%
			f6	6.3	98.3%	100%

6.5%, and precision rate by 3.6%, showing that DRZ use enhanced typing speed and accuracy.

3. Dwell Time (tp): Longer dwell times ($t2$) increased time per character by 10.5%, while improving precision by 8.3%, suggesting enhanced accuracy but slower typing.

Additional analyses showed that taller volunteers had slower typing speeds and lower precision rates compared to shorter volunteers. However, no meaningful differences were observed in terms of typing experience.

After the experiment, volunteers rated the system 10/10 for its effectiveness in assisting individuals with motor disabilities in communication. The assisted virtual keyboard also received a 10/10 rating for meeting expectations and supporting user interaction. Usability was rated 9/10, and system performance (accuracy and response time) was rated 8/10, despite some reports of discomfort due to prolonged use. These results are consistent with the findings of Attiah and Khairullah (2021), in which the system was rated 10/10 in importance and 8/10 in overall quality, although 5/10 participants believed that eye blinking might not be the most effective interaction method. The data reinforce the effectiveness of the developed solution, highlighting its potential as a more accurate, intuitive, and comfortable alternative.

V. CONCLUSION

Interaction with computers via keyboard and mouse can pose challenges for individuals with motor impairments. Although digital AAC devices, such as eye trackers, promote inclusive communication, they are often expensive and inaccessible [8]. This study proposes a low-cost, eye-assisted virtual keyboard guided by gaze and head orientation, employing real-time facial recognition and a CNN for pointer prediction, along with a language model for word suggestions. The evaluation focused on four key aspects: facial recognition accuracy, CNN prediction accuracy (measured by RMSE), recommendation system performance, and user experience based on volunteer testing

Facial recognition achieved a 99.9% success rate on the Eye Chimera dataset. The CNN results showed the best performance with the multiple inputs model, yielding an RMSE of 31.58 pixels. For the LM, the trigram outperformed the unigram and bigram models, achieving a perplexity of 10.0 with a 91.1% improvement over the unigram. The volunteer tests showed an average time per character of 7.3 seconds, a precision rate of 93%, and a 100% success rate. Qualitative feedback indicated high usability (9/10) and accuracy (8/10) ratings.

Future work will focus on enhancing the assisted keyboard interface, improving the word recommendation system, expanding volunteer testing, and comparing additional prediction models. The interface will incorporate input through blinking and alternative keyboards (e.g., alphanumeric, non-QWERTY, adaptive). The word recommendation system will be improved to handle corrections for partially or previously typed words, thereby enabling further evaluation of its impact on user performance. Testing will be expanded to include more sentences, additional volunteers, and individuals with motor disabilities, thereby increasing the representativeness of the sample. A comparison with other models will evaluate the performance of the proposed model and its impact on RMSE.

ACKNOWLEDGMENTS

The authors would like to thank FEI, the Ethics Committee, and the volunteers for their support.

REFERENCES

[1] H. Wang, D. Yu, Y. Zeng, T. Zhou, W. Wang, X. Liu, Z. Pei, Y. Yu, C. Wang, Y. Deng, and A. Cheshmehzangi, "Quantifying the impacts

of posture changes on office worker productivity: an exploratory study using effective computer interactions as a real-time indicator," *BMC Public Health*, vol. 23, 2023. DOI: 10.1186/s12889-023-17100-w.

[2] R. A. da Silva and A. C. Paschoarelli Veiga, "Algorithm for decoding visual gestures for an assistive virtual keyboard," *IEEE Latin America Transactions*, vol. 18, p. 1909–1916, Mar. 2021. DOI: 10.1109/TLA.2020.9398632.

[3] D. D. Aoife McNicholl, Hannah Casey and P. Gallagher, "The impact of assistive technology use for students with disabilities in higher education: a systematic review," *Disability and Rehabilitation: Assistive Technology*, vol. 16, no. 2, pp. 130–143, 2021. DOI: 10.1080/17483107.2019.1642395.

[4] F. L. Silva and A. R. C. Serra, "Tecnologia assistiva: recursos de comunicação aumentativa e alternativa na proposta de interação e aprendizagem dos alunos com autismo," *Revista Tempos e Espaços em Educação*, 2023. DOI: 10.20952/revtee.v16i35.18610.

[5] L. F. B. Loja *et al.*, "Tecnologia assistiva: um teclado virtual evolutivo para aplicações em sistemas de comunicação alternativa e aumentativa," 2015. DOI: 10.14393/ufu.te.2015.153.

[6] A. C. A. Montenegro, L. K. S. de Melo Silva, R. C. de Sá Bonotto, R. A. Lima, and I. A. de Lavor Navarro Xavier, "Uso de sistema robusto de comunicação alternativa no transtorno do espectro do autismo: relato de caso," *Revista CEFAC*, 2022. DOI: 10.1590/1982-0216/202224211421s.

[7] C. M. Togashi and C. C. de Figueiredo Walter, "As contribuições do uso da comunicação alternativa no processo de inclusão escolar de um aluno com transtorno do espectro do autismo," 2016. DOI: 10.1590/S1413-65382216000300004.

[8] R. Bonotto, Y. Corrêa, E. Cardoso, and D. S. Martins, "Oportunidades de aprendizagem com apoio da comunicação aumentativa e alternativa em tempos de covid-19," *Revista Ibero-Americana de Estudos em Educação*, 2020. DOI: 10.21723/riaee.v15i4.13945.

[9] G. K. dos Santos Silva, "A comunicação alternativa como aporte para inclusão," *Revista Ibero-Americana de Humanidades, Ciências e Educação*, 2023. DOI: 10.51891/rease.v9i4.9253.

[10] L. dos Santos Batista, K. M. O. Kumada, and P. Benitez, "Rastreamento ocular e a educação especial inclusiva: uma revisão sistemática," *Olhar de Professor*, 2023. DOI: 10.5212/OlharProfr.v.26.19672.002.

[11] M. Nazar, M. M. Alam, E. Yafi, and M. M. Su'ud, "A systematic review of human–computer interaction and explainable artificial intelligence in healthcare with artificial intelligence techniques," *IEEE Access*, vol. 9, pp. 153316–153348, 2021. DOI: 10.1109/ACCESS.2021.3127881.

[12] D. Freitas, S. Rodrigues, and J. Ribeiro, "Interfaces de acesso ao computador para pessoas com limitações motoras: um estado da arte," *Tecnologias assistivas: formação, experiências e práticas*, pp. 156–175, 2024. DOI: 10.52695/978-65-5456-050-4.8.

[13] J. Hori, K. Sakano, and Y. Saitoh, "Development of communication supporting device controlled by eye movements and voluntary eye blink," in *The 26th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, vol. 2, pp. 4302–4305, 2004. DOI: 10.1109/IEMBS.2004.1404198.

[14] J. Wobbrock, J. Rubinstein, M. Sawyer, and A. Duchowski, "Not typing but writing: Eye-based text entry using letter-like gestures," 01 2007. DOI: 10.1145/1344471.1344475.

[15] A. Bulling, D. Roggen, and G. Tröster, "It's in your eyes: Towards context-awareness and mobile hci using wearable eog goggles," vol. 344, pp. 84–93, 01 2008. DOI: 10.1145/1409635.1409647.

[16] N. Bee, Nikolaus, E. Andre, and Elisabeth, "Writing with your eye: A dwell time free writing system adapted to the nature of human eye gaze," 06 2008. DOI: 10.1007/978-3-540-69369-7_13.

[17] P. Majaranta, U.-K. Ahola, and O. Špakov, "Fast gaze typing with an adjustable dwell time," pp. 357–360, 04 2009. DOI: 10.1145/1518701.1518758.

[18] W. Tangsuksant, C. Aekmunkhongpaisal, P. Cambua, T. Charoenpong, and T. Chanwimalueang, "Directional eye movement detection system for virtual keyboard controller," in *The 5th 2012 Biomedical Engineering International Conference*, pp. 1–5, 2012. DOI: 10.1109/BMEiCon.2012.6465432.

[19] V. I. Saraswati, R. Sigit, and T. Harsono, "Eye gaze system to operate virtual keyboard," in *2016 International Electronics Symposium (IES)*, pp. 175–179, 2016. DOI: 10.1109/ELECSYM.2016.7860997.

[20] H. Cecotti, Y. K. Meena, and G. Prasad, "A multimodal virtual keyboard using eye-tracking and hand gesture detection," in *2018 40th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pp. 3330–3333, 2018. DOI: 10.1109/EMBC.2018.8512909.

[21] S. Tantisatirapong and M. Phothisonothai, "Design of user-friendly virtual thai keyboard based on eye-tracking controlled system," in *2018 18th International Symposium on Communications and Information Technologies (ISCIT)*, pp. 359–362, 2018. DOI: 10.1109/ISCIT.2018.8587965.

[22] M. I. Rusydi, A. Anandika, R. Adnan, K. Matsuhita, and M. Sasaki, "Adaptive symmetrical virtual keyboard based on eog signal," in *2019 4th Asia-Pacific Conference on Intelligent Robot Systems (ACIRS)*, pp. 22–26, 2019. DOI: 10.1109/ACIRS.2019.8935956.

[23] A. Z. Attiah and E. F. Khairullah, "Eye-blink detection system for virtual keyboard," in *2021 National Computing Colleges Conference (NCCC)*, pp. 1–6, 2021. DOI: 10.1109/NCCC49330.2021.9428797.

[24] A. Anandika, M. I. Rusydi, P. P. Utami, R. Hadelina, and M. Sasaki, "Hand gesture to control virtual keyboard using neural network," *JITCE (Journal of Information Technology and Computer Engineering)*, 2023. DOI: 10.25077/jitce.7.01.40-48.2023.

[25] M. Shaima, N. Nabi, M. N. U. Rana, E. Ahmed, M. I. Tusher, M. Hasan, Mukti, and Q. Saad-Ul-Mosaher, "Elon musk's neuralink brain chip: A review on 'brain-reading' device," *Journal of Computer Science and Technology Studies*, 2024. DOI: 10.32996/jcsts.

[26] H. Drewes, "Eye gaze tracking for human computer interaction," 03 2010. DOI: 10.5282/edoc.11591.

[27] K. H. et al., "Eye tracking: empirical foundations for a minimal reporting guideline," *Behavior Research Methods*, vol. 55, pp. 364 – 416, 2022. DOI: 10.3758/s13428-021-01762-8.

[28] D. E. King, "Max-margin object detection," *ArXiv*, vol. abs/1502.00046, 2015. DOI: 10.48550/arXiv.1502.00046.

[29] D. da Silva Lima, *Avaliação da função visual infantil a partir de solução automatizada de rastreamento ocular baseada em vídeo*. PhD thesis, Instituto de Psicologia, University of São Paulo, 2024. DOI: 10.11606/T.47.2024.tde-13062024-113449.

[30] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Adaptive computation and machine learning, MIT Press, 2016. DOI: 10.1007/s10710-017-9314-z.

[31] J. Maher, *The Future Was Here: The Commodore Amiga*. Cambridge, MA: MIT Press, 2012. DOI: 10.7551/mitpress/9022.001.0001.

[32] N. Garay-Vitoria and J. Abascal, "Text prediction systems: A survey," *Universal Access in the Information Society*, vol. 4, no. 3, pp. 188–203, 2006. DOI: 10.1007/s10209-005-0005-9.

[33] L. Florea, C. Florea, R. Vranceanu, and C. Vertan, "Can your eyes tell me how you think - a gaze directed estimation of the mental activity," pp. 60.1–60.11, 01 2013. DOI: 10.5244/C.27.60.

[34] L. Li, K. Jamieson, A. Rostamizadeh, E. Gonina, J. Ben-tzur, M. Hardt, B. Recht, and A. Talwalkar, "A system for massively parallel hyperparameter tuning," in *Proceedings of Machine Learning and Systems* (I. Dhillon, D. Papailiopoulos, and V. Sze, eds.), vol. 2, pp. 230–246, 2020. DOI: 10.48550/arXiv.1810.05934.

[35] J. Yacim and D. Boshoff, "Impact of artificial neural networks training algorithms on accurate prediction of property values," *Journal of Real Estate Research*, vol. 40, pp. 375–418, 11 2018. DOI: 10.1080/10835547.2018.12091505.

Bruno Waideman has been working at Itaú as a Data Manager since 2020. He is involved in building classification and prediction models. Bruno holds a Bachelor's degree in Computer Science from FEI and a Postgraduate degree in Finance and Economics from FGV. Holds a Master's degree Artificial Intelligence from FEI.

Plínio Thomaz Aquino Jr. holds a Bachelor's degree in Computer Science (1998) and a Master's degree in Software Engineering (2001) from UFSCar, the latter developed in collaboration with the German National Research Center for Information Technology. He completed his Ph.D. in Electrical Engineering at EP-USP in 2008. Has been a lecturer at FEI, where he supervises Master's and Ph.D. students in the Artificial Intelligence. His expertise includes Human-Machine Interaction, Robotics, IoT, Software Engineering and Autonomous Robotics.