LookASD – Intelligent System to Assist Healthcare Professionals in Decision-Making About Children with ASD

Jarriv Alcantara de Oliveira Reginaldo
Francisco de Assis Zampirolli
Wagner Tanaka Botelho

[1]Federal University of ABC (UFABC) – Santo André – SP – Brazil

`{jarriv.reginaldo,fzampirolli,wagner.tanaka}@ufabc.edu.br`

Abstract. *Autism Spectrum Disorder (ASD) is characterized by developmental neurological disorders that impact individuals' social skills, communication, and learning abilities. It affects approximately one in 200 people. People with ASD are born with the disorder, and there is no cure. However, simple and reliable methods that utilize public data still need to be made available to assist healthcare professionals in diagnosing ASD. This paper presents* LookASD*, an open-source system that displays multiple images to capture the visual attention patterns (focal maps) of children ages five to 12. It uses a webcam to capture children's focal maps and then classifies them based on a public dataset of focal maps collected from children with and without ASD. Thirty-four machine-learning classification methods from the Sklearn library in Python were applied to this dataset, with 80% for training and 20% for testing. The most accurate classifiers were Support Vector Classification (SVC), K-Nearest Neighbors (KNeighborsClassifier), and Histogram Gradient Boosting (HistGradientBoostingClassifier).*

Resumo. *O Transtorno do Espectro Autista (TEA) é caracterizado por distúrbios neurodesenvolvimentais que afetam as habilidades sociais, a comunicação e as capacidades de aprendizado das pessoas. Afeta aproximadamente uma em cada 200 pessoas. As pessoas com TEA nascem com o transtorno e não há cura. No entanto, ainda é necessário disponibilizar métodos simples e confiáveis que utilizem dados públicos para auxiliar os profissionais de saúde no seu diagnóstico. Este artigo apresenta o LookASD, um sistema de código aberto que exibe múltiplas imagens para capturar os padrões de atenção visual (mapas focais) de crianças de cinco a 12 anos. Para isso, uma webcam captura os mapas focais e, em seguida, os classificam usando um conjunto de dados públicos composto por mapas focais de crianças com e sem TEA. Na classificação, trinta e quatro métodos de aprendizado de máquina da biblioteca Sklearn foram utilizados no conjunto de dados, sendo 80% para treinamento e 20% para teste. Os classificadores mais precisos foram a Classificação de Vetores de Suporte (SVC), K-Vizinhos Mais Próximos (KNeighborsClassifier) e o Histograma Gradient Boosting (HistGradientBoostingClassifier).*

1. Introduction

Autism Spectrum Disorder (ASD) is a neurodevelopmental disorder that affects approximately one in every 200 people [Klin 2006]. People with ASD have difficulties in socializing, communicating, and learning. Additionally, repetitive and restrictive behavior patterns are also present. Although there is no cure, and the cause is still unknown, scientific advances have allowed for a better understanding of the disorder [Klin 2006].

The diagnosis of ASD remains complex and widely debated since the characteristics of a person with ASD have changed over time. Before World War I, Child Psychology considered ASD a psychological disorder without conclusive data for diagnosis. According to Donovan and Zucker [Donvan and Zucker 2017], ASD has existed since the dawn of humanity. However, due to inattention or lack of interest, it has often been misdiagnosed, thereby impeding treatment.

Recent studies have shown that children diagnosed with autism present anomalies in different areas between 12 and 18 months of age, such as atypical visual interests, delays in motor acquisition, limitations in play, difficulties in social communication and language, as well as difficulties in learning daily activities [Oliveira 2009]. Furthermore, a systematic review of visual processing in ASD has been presented by Simmons et al. [Simmons et al. 2009]. As a result, individuals with ASD tend to focus on distinct parts of an image when viewing a scene with individuals and landscapes.

This paper presents a web-based system known as *LookASD*, developed to assist healthcare professionals in diagnosing ASD in children aged five to 12 years. The system utilizes a public dataset created by Duan et al. [Duan et al. 2019]. It has 300 images of faces, people, landscapes, animals, plants, and other objects. Each image was presented to 14 children with ASD and 14 typically developing children. Their eye gaze patterns were captured using an eye-tracker (Tobii T120) to generate a "focal map" of the image. The *LookASD* uses this focal map to capture real-time gaze patterns of children using a webcam and applies classifiers from the Sklearn library (`scikit-learn.org`) to predict whether the child is likely to have ASD or not. The analysis revealed that the Top-Performing Classifiers are the SVC, KNeighborsClassifier, and HistGradientBoostingClassifier.

As a contribution, *LookASD* is the only system found in the literature that can be accessed anywhere. It utilizes real gaze data to predict the likelihood of children having ASD. It is important to highlight that the purpose of this paper is to present how the system was implemented. Thus, ethical approval from a research committee was not required as the dataset was created by Duan et al. [Duan et al. 2019], and human experts have not yet validated the *LookASD*.

This paper presents the development of *LookASD*, a system that uses real gaze data to predict the likelihood of children having ASD. The literature review in Section 2 includes the classification of ASD, examples of classifiers, and related work. In Section 3, the materials and methods section describes the proposed system, use case diagram, flowcharts, and database. Results in Section 4 show that the system achieved an average accuracy of 81% in identifying children with ASD. Finally, in Section 5, the conclusions highlight the potential of *LookASD* as a valuable tool to assist health professionals.

2. Literature Review

This section introduces the main theoretical concepts related to data classification, showcasing examples of four widely used techniques: Support Vector Machines (SVM), K–Nearest neighbor, Decision Trees, Artificial Neural Networks (ANNs), and Ensemble Method. Additionally, notable related works in the field are discussed.

2.1. Classification

Classification is a supervised Machine-Learning (ML) method that involves creating a model, called a classifier, capable of assigning labels to the classes of new unknown data [Quilici-Gonzalez and Zampirolli 2014]. Information is a set of data that has value in a specific context, while data is information without initial usefulness, such as the pixels of an image. For example, in a gender recognition system, the information would be the class (or label) associated with an image of a person.

According to [Han and Kamber 2006], classification is divided into two steps:

Training: a large portion of the dataset is used to "teach" the classifier to label the classes correctly. Typically, 70-80% of the data is used to create the classifier model;

Testing: around 20-30% of the data used in the training step to verify the classifier's effectiveness. The goal is to ensure the classifier can classify new examples without defined classes.

The division of data into training and testing sets reduces the available samples for the model, rendering the results reliant on random selection. Cross-validation, such as the k-fold method, addresses this issue [Ojala and Garriga 2010]. Therefore, training occurs on subsets of training data, while evaluation happens on the remaining data. The final metric is the average of metrics obtained in each iteration. Despite resource requirements, this approach minimizes data wastage, proving advantageous when sample size is limited.

Finally, classification parameters such as accuracy are used to evaluate the classifier's effectiveness.

2.2. Classifiers

Many classifier examples are available in the Sklearn (`scikit-learn.org`), such as:

Support Vector Machines

SVM is a ML technique based on concepts from the Statistical Learning theory [Vapnik 1999]. It is a supervised ML algorithm used for classification and regression purposes. SVM maps input data into a higher-dimensional space and finds a hyperplane that best separates the data classes. This hyperplane is chosen to maximize the margin between the classes, which is the distance between the hyperplane and the closest points of each class.

SVM is popular due to its effectiveness in handling high-dimensional datasets and separating non-linear data classes. It is also known for its ability to generalize to unseen data, making it useful in real-world applications.

K–Nearest Neighbor

Nearest neighbor classifiers are based on learning by analogy: comparing a given test dataset with similar training datasets. Training datasets are described by n number of attributes. Each dataset represents a point in an $n-$dimensional space. Therefore, all training datasets are stored in an $n-$dimensional space. When given an unknown dataset, a k-nearest neighbor classifier searches the pattern space for the k closest training datasets to the unknown dataset. These k training datasets are the k nearest neighbors of the unknown dataset [Han and Kamber 2006].

Decision Tree

In the decision tree, nodes are created and connected by edges, starting at the root node at the top of the structure. Next, decision nodes are added, with the results represented by branches until the leaf nodes. A branch can lead to a new decision node or a final leaf node, which means the classifier's final decision [Larose 2015].

Decision trees require certain requirements to work correctly: training datasets must be provided for the algorithm to learn how to construct the decision tree; the provided data must be varied to optimize the algorithm's performance; and the target attribute classes must be discrete. In addition, as an example that uses decision trees in autism research, [Usta et al. 2019] analyzed several predictive factors that may influence a child's likelihood of having autism.

Ensemble Method

The ensemble method combines k learning models, also known as base classifiers $M_1, M_2, \cdots, M_k$, to create an improved classification model called M^*. To achieve this, dataset D creates k training sets $D_1, D_2, \cdots, D_k$, where D_i, $1 \leq i \leq k-1$, generates the classifier M_i. When a new dataset is received for classification, each base classifier emits a class prediction through voting. The ensemble method then returns a class prediction based on the votes of the base classifiers [Han and Kamber 2006]. The `HistGradientBoostingClassifier` is an ensemble method in Sklearn. Specifically, it is an implementation of the historical gradient boosting algorithm, a type of ensemble algorithm combining multiple decision tree models to improve classification accuracy [Cournapeau 2023].

2.3. Related Developed Systems

The unique behavioral characteristics and interests of people with ASD offer potential for multiple research studies. Table 1 compares the *LookASD* system with its main features with other related systems proposed by [Nogueira Filho 2020, Biasão 2019, Pal and Rubini 2022, Bortoletti 2022].

Nogueira Filho [Nogueira Filho 2020] developed a system to assist in detecting ASD using ML and eye-tracking. For this, children's behaviour was analyzed when viewing superhero images. The author highlighted that the unsupervised prediction model demonstrated high accuracy in predicting the behaviour of children with ASD. On the other hand, [Biasão 2019] captures the focal map of individuals with ASD, aged between 3 and 16 years. ML techniques are applied to label the severe and non-severe ASD groups through visual areas of interest and the Childhood Autism Rating Scale (CARS) classification. Once the classifier is trained, it predicts the presence and severity of ASD in individuals.

Table 1. Comparative Analysis of Related Works.

Paper	Object of study	Data acquisition	Online	Dataset [Duan et al. 2019]
[Nogueira Filho 2020]	Visual	*Eye-Tracker*/PC	No	No
[Pal and Rubini 2022]	Facial Expressions/Sounds	*Webcam*/Microphone/PC	No	No
[Bortoletti 2022]	Facial Expressions	*Webcam*/Android	Yes	No
[Biasão 2019]	Visual	*Eye-Tracker*/PC	No	No
LookASD	Visual	*Webcam*/PC	Yes	Yes

Pal and Rubini [Pal and Rubini 2022] implemented a system to assist healthcare professionals in diagnosing ASD using facial expressions and children's speech sounds. However, this approach is particularly challenging because children with ASD often have difficulty expressing emotions through their faces and speaking. [Bortoletti 2022] developed a smartphone application to record a video of a human face. From the video, ML techniques were used to analyze facial features to diagnose ASD.

With the information about the applications developed by [Nogueira Filho 2020, Biasão 2019, Pal and Rubini 2022, Bortoletti 2022], and considering Table 1, it is possible to verify that all works used devices to capture images, such as webcam and eye-tracker. Webcam is low-cost, and notebooks already have this device. However, eye-trackers are expensive. In addition, only [Bortoletti 2022] developed an application for Android smartphones that can be accessed via the Internet, and [Pal and Rubini 2022] used a microphone to capture the children's speech sounds. On the other hand, the *LookASD* is a web-based system that uses a public dataset created by [Duan et al. 2019], allowing the evaluation to be carried out immediately after use. Unlike [Bortoletti 2022], it does not record videos of the evaluated individuals and does not need to share them for later evaluation. However, [Bortoletti 2022], and the *LookASD* system requires accurate facial detection of the evaluated individual to work correctly. [Biasão 2019] built its database, and it aims to identify the level of ASD in individuals who have already been diagnosed. In contrast, the *LookASD* system aims to analyze whether children are with and without ASD. Despite these differences, [Biasão 2019] and the *LookASD* use classifier techniques to label the collected information, and the tools present images to capture the focal maps.

3. Materials and Methods

This section presents an overview of the *LookASD* system. The use case and flowchart diagrams are described. Finally, the *LookASD* interface and the dataset provided by [Duan et al. 2019] are presented.

3.1. *LookASD* System Operation

Figure 1 shows how the *LookASD* system works. In the first step, the (a) User is represented by a child between 5 and 12 years old and is assisted by an (b) Adult while using the system. The (c) Equipment connected to the internet is a computer desktop or notebook with a webcam. In the Second Step, all the tools and libraries needed to run the *LookASD* are installed on the server.

In Figure 1, the (a) User sits in front of the (c) Equipment in a well-lighted environment. After that, the webcam captures the x and y coordinates, known as the focal map, related to the gaze data of the (a) User when viewing the image displayed on the monitor.

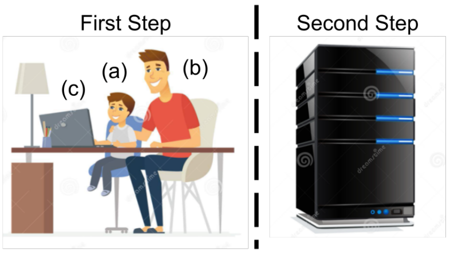


Figure 1. Operation of the *LookASD* System.

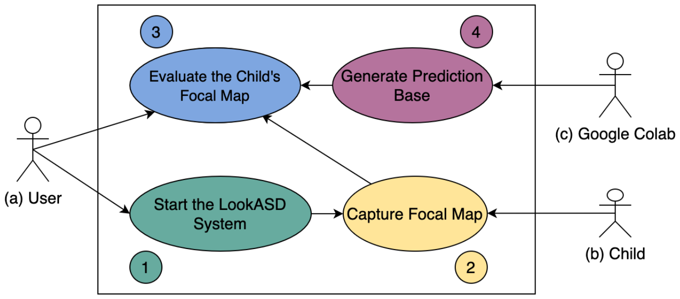


Figure 2. Use Case Diagram.

After capturing the focal map, the data is processed in the Second Step to predict whether the child is likely to have ASD or not.

In conclusion, the *LookASD* system is a simple and accessible tool that uses a webcam to capture the child's focal map and predict the points, providing valuable information for healthcare professionals to help the ASD diagnosis.

Use Case Diagram

Figure 2 illustrates the actors and actions implemented in the *LookASD* system, described in Section 3.1, through the use case diagram. The actor (a) User executes the use cases ① Start the *LookASD* and ③ Evaluate the Child's Focal Map. The actor (b) Child captures data in the ② Capture Focal Map. The use case ④ Generate Prediction Base, which uses the processed dataset by the actor (c) Google Colab (`colab.research.google.com`), forms the prediction base for ③ to analyze the focal map obtained in ②, indicating whether or not the actor (b) is on the ASD. In addition, the flowchart in Figure 3 details the behavior of each use case.

Flowchart

The flowchart in Figure 3 follows the same color scheme and numbering as in Figure 2 for a better understanding of how the *LookASD* works. In the use case ①, the *LookASD* system runs (1-Start) in a web browser, as shown in Figure 4, to capture the focal map, which prompts the user to (①-2) Enter Personal Information through a text box described in Section 3.2 and grants (①-3) Permission to Use the Webcam. If permission is granted in (①-3), the User's face shape is detected in (①-4). After that, the User's facial view is displayed in Figure 4 (a) using the WebGazer [Papoutsaki et al. 2016] in (①-5) to calibrate the focal map points captured by the webcam in (①-6). The calibration is performed at a distance of 60 cm between the user's eyes and the screen.

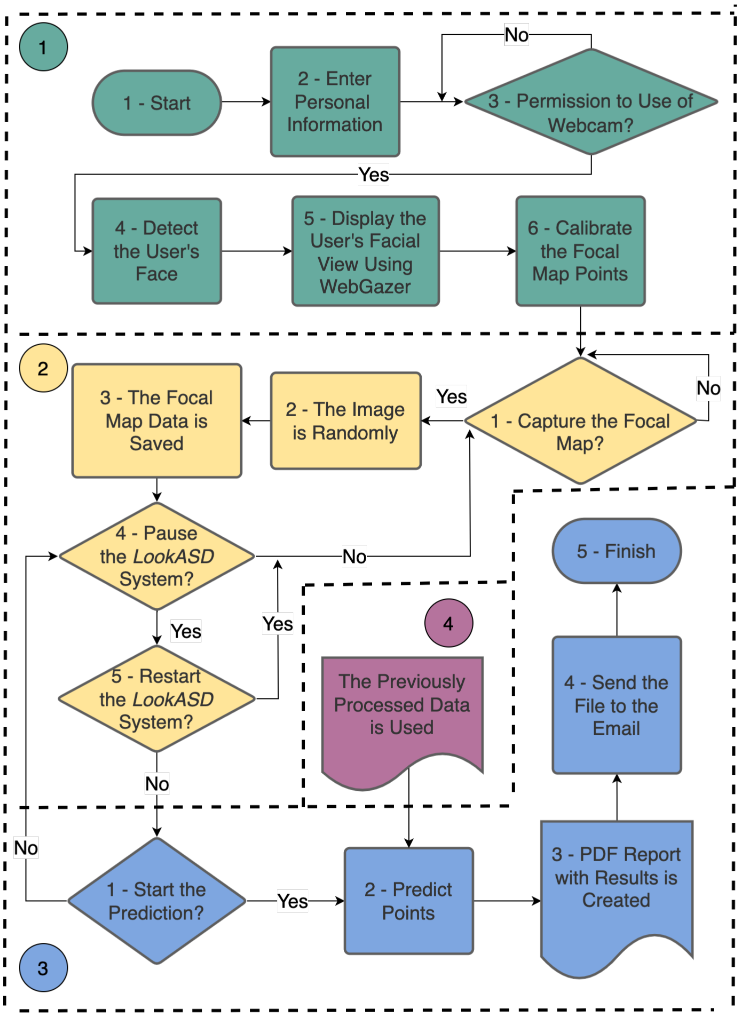


Figure 3. Flowchart of the First and Second Steps of the *LookASD* System in Figure 1.

After the initial setup in Figure 3 ①, the next step in (2-1) is performed to capture the focal map data for analysis. Therefore, a dataset image, presented in Figure 4 (b), is randomly selected in (2-2). It is important to highlight that the chosen (b) image is not selected again. In addition, the child views the (b) image on the screen for three seconds, and every half-second, the (c) points of the focal map data are saved in a TXT file, as shown in (2-3). However, if the (2-4) (d) Pause button is pressed, the *LookASD* system is paused, restarted in (2-5), or starts the prediction in ③ when the action (3-1) (e) Evaluate button is pressed. After that, the action (3-2) Predict Points is executed, using the previously processed data in ④. In this action, a PDF report of the prediction results, illustrated in Figure 5, is generated in (3-3), and the file is sent to the email provided at the beginning of the focal map capture in (3-4). Finally, the *LookASD* is finished in (3-5).

The use case ④ Generate Prediction Base in Figures 2 and 3 is detailed in Figure 6. The Google Colab is initialized in ①, and the needed libraries to run the *LookASD* system are loaded in ②. Then, the images from dataset ③ are loaded, and the Sklearn (`scikit-learn.org`) classifiers are used in ④. After that, the classifiers are tested for each of the 300 images in ⑤, the best results are saved in a TXT file ⑥, and the system is finished in ⑦.

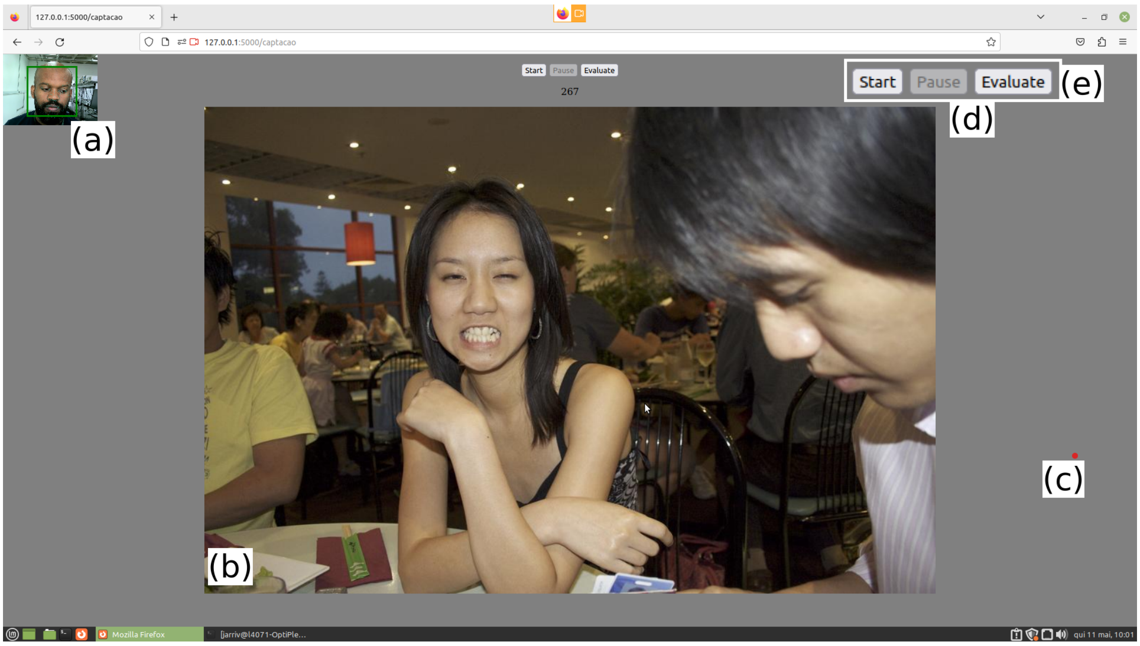
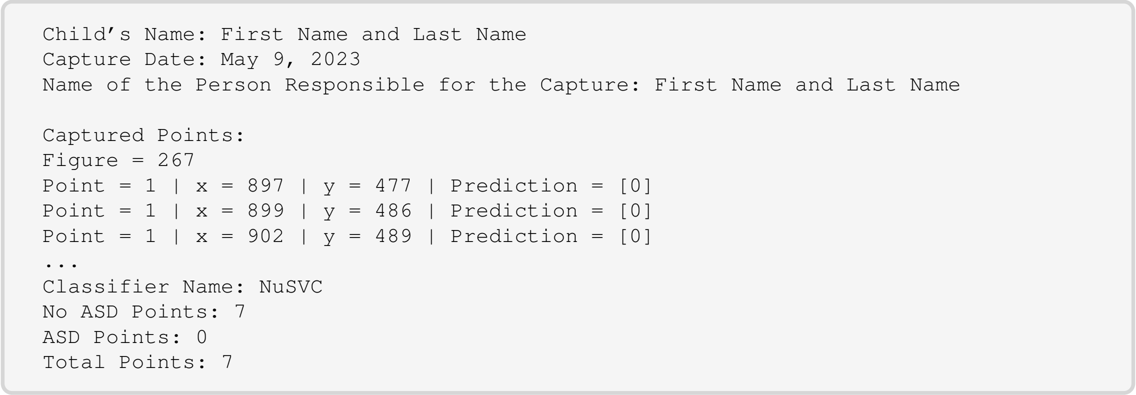


Figure 4. The *LookASD* System Interface.



```
Child's Name: First Name and Last Name
Capture Date: May 9, 2023
Name of the Person Responsible for the Capture: First Name and Last Name

Captured Points:
Figure = 267
Point = 1 | x = 897 | y = 477 | Prediction = [0]
Point = 1 | x = 899 | y = 486 | Prediction = [0]
Point = 1 | x = 902 | y = 489 | Prediction = [0]
...
Classifier Name: NuSVC
No ASD Points: 7
ASD Points: 0
Total Points: 7
```

Figure 5. A PDF Report of the Prediction Results.

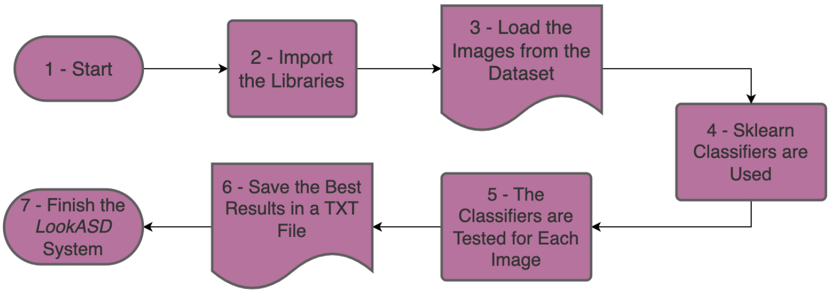


Figure 6. Google Colab Operation Flowchart.

3.2. *LookASD* System Description

The *LookASD* system was implemented on a Lenovo G400s notebook with 4GB of RAM, running the Linux Mint 20.2 operating system, and a screen resolution of 1366×768 with a refresh rate of 60 Hz. Focal map points were captured using a Logitech C920 webcam with Full High Definition (HD).

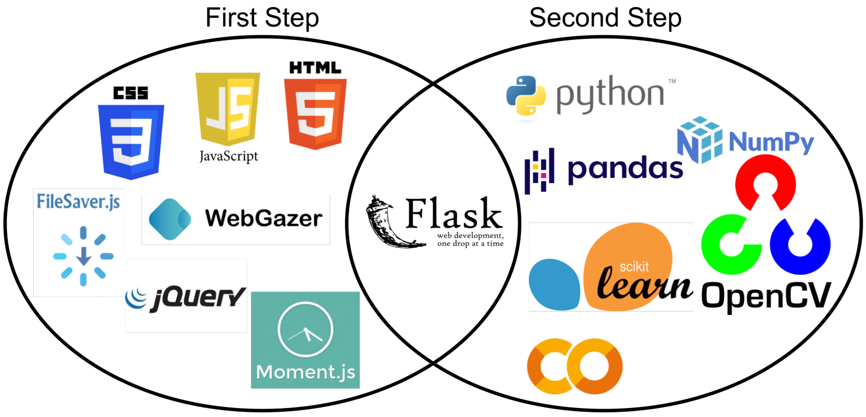


Figure 7. Languages, Libraries, and Tools Used to Develop *LookASD* System.

Languages, Libraries, and Tools Used in the Development

Figure 7 presents the languages, libraries, and tools used to develop the system and highlights that Flask (`flask.palletsprojects.com`) is responsible for information exchange between the First and Second Steps in Figure 1.

In the First Step of Figures 1 and 7, Cascading Style Sheet (CSS) is used with HyperText Markup Language (HTML) to ensure the resizing of the images presented on the *LookASD* system interface of Figure 4(b). Moment.js handles dates in JavaScript, and jQuery gets the width and height of images shown in the *LookASD* system interface. WebGazer.js is a library to track the user's eyes through a webcam in real time and infers the eye-gaze location in the web browser. After that, the focal map points are saved in a TXT file using JavaScript and FileSaver.js. jsGlue, integrated with Flask, transmits the focal map data to the server in the Second Step of Figure 1.

In Figures 1 and 7, related to the Second Step, Python scripts are implemented in Google Colab to identify the focal map points captured in the First Step and processed in the Server of Figure 1. For this, some libraries are used. NumPy calculates the focal map points regardless of monitor resolution, and Pandas opens the TXT file created in the First Step to analyse the data. OpenCV draws the points on the dataset images used to create the child's focal map to visualise the results better. In addition, on each image in the dataset, ten cross-validation tests are performed for each of the 34 Sklearn classifiers. After that, the classifier with the best average is associated with an image and saved in a TXT file that also contains the width and height of each image.

Dataset

The dataset used in this work was created by [Duan et al. 2019] using a Tobii T120 Eye-Tracker with a resolution of 1280×1024, a sampling rate of 60Hz, and a capture distance of 65cm. It has 300 images used in 10 sessions to capture the focal map of 14 children with ASD and 14 without ADS. For this, images of people, animals, and others were presented randomly, with a viewing time of three seconds and a one-second interval between them. However, data from six children were not used due to the difficulty in calibrating the system.

(a) 267 (b) (c)

(d) 254 (e) (f)

(g) 41 (h) (i)

(j) 220 (k) (l)

Figure 8. Images with their Focal Maps of Children with and without ASD.

Focal Maps

Figure 8 shows Images (a) 267, (d) 254, (g) 41 and (j) 220 and the focal maps represented by (b-c), (e-f), (h-i) and (k-l), respectively. The results presented in Section 4 ranked Images (a) and (d) as the best performing, while Images (g) and (j) were considered the worst performing. It is important to highlight that the images are labelled with their identification numbers in the dataset. The focal maps in (b-e-h-k) and (c-f-i-l) display the regions most relevant to classifying children with and without ASD, respectively. This information helps to understand the visual features that contribute to the classification task and for developing algorithms to improve the accuracy of ASD detection in children. It is noticeable that the focal map points for children with ASD are more dispersed in (b) and (e), compared to (c) and (f), related to children without ASD. Therefore, these differences suggest that children with ASD may have difficulty fixing their gaze on objects like faces.

In addition, the focal maps in (k) and (l) are similar and represent a scene with an animal in (j). It suggests that making accurate predictions using the method presented in this Section 3 is impossible. However, in the group of people from Image 41, the same focal map is also present for children with and without ASD.

4. Results

This section presents the results of the performance metrics obtained from the 34 Sklearn classifiers applied to the 300 images of the dataset, using 80% of the data for training and 20% for testing. All classifiers were used with their default parameter values.

4.1. Metrics for the Top-Performing Classifiers

Table 2 displays the performance metrics for the two best and two worst images, shown in Figure 8 (a) 267 – (d) 254, and (g) 41 – (j) 220, respectively. The Confusion Matrix lines display the results of each classifier for the corresponding image. For instance, the NuSVC classifier obtained the following information for Image 267: TN - correctly predicted 52 negative instances, FP - incorrectly predicted two positive instances, FN - did not predict seven positive instances that were positive, and TP - correctly predicted ten positive instances. In addition, the evaluation metrics for each classifier and image are Accuracy, Sensibility (ability to predict positives - TP Rate), Specificity (ability to predict negatives - TN Rate), Precision (of the predicted positives, how many were positive), Root Mean Square Error (RMSE), Mean Absolute Error (MAE) and F1-Score (a metric that considers Precision and Sensitivity). In summary, achieving high rates of TP and TN and low rates of FP and FN is desirable. Precision measures the proportion of TPs among the predicted positive cases, while sensitivity determines the proportion of TPs among all actual positive cases. The F1-Score calculates the harmonic mean between Precision and Sensitivity, providing a measure of the accuracy reliability. It relates these two metrics and identifies possible system failures when the score approaches zero [Quilici-Gonzalez and Zampirolli 2014].

Table 2 shows that the NuSVC was the top-performing classifier for Images 267 and 41, while the KNeighborsClassifier achieved the best results for Image 254. The BernoulliNB classifier was the top performer for Image 220. The Accuracy, Sensitivity, Specificity, and Precision show a decrease in values from Image 267 to Image 220, except for the Specificity metric of the worst Image 220, which obtained a value of 100% due to an FP value being zero. The RMSE and MAE metrics showed lower results for Image 267, increasing for Image 220. The F1 Score was high for Image 267 but decreased for Image 220, with a critical case found for the worst image, which obtained a value of 0 due to FP and TP values being zero.

ROC Curves

A Receiver Operating Characteristic (ROC) [Gribskov and Robinson 1996] curve is a graph that shows how well a binary classifier system can distinguish between positive and negative samples. It plots the true positive rate against the false positive rate at different threshold settings. The curve helps to compare the performance of different classifiers and evaluate their overall performance. The higher AUC values indicate better performance.

Figure 9 displays the ROC curves of the best (267 and 254) and worst (41 and 220) Images, shown in Figure 8, analyzed by the Sklearn classifiers. Images 267 and 254

Table 2. Metrics for the Top-Performing Classifiers on the Two Best and Two Worst Images in the Dataset.

		Images			
		267	**254**	**41**	**220**
Best Classifiers		*NuSVC*	*KNeighborsClassifier*	*NuSVC*	*BernoulliNB*
Confusion Matrix	True Negative/TN	52	40	35	36
	False Positive/FP	2	2	18	0
	False Negative/FN	7	10	27	28
	True Positive/TP	10	10	31	0
Metrics	Accuracy	87,32%	80,65%	59,46%	56,25%
	Sensibility	58,52%	50%	53,45%	0%
	Specificity	96,3%	95,24%	66,04%	100%
	Precision	83,33%	83,33%	63,27%	0%
	Root Mean Square Error/RMSE	0,36	0,44	0,64	0,66
	Mean Absolute Root/MAR	0,13	0,19	0,41	0,44
	F1 Score	0,69	0,62	0,58	0

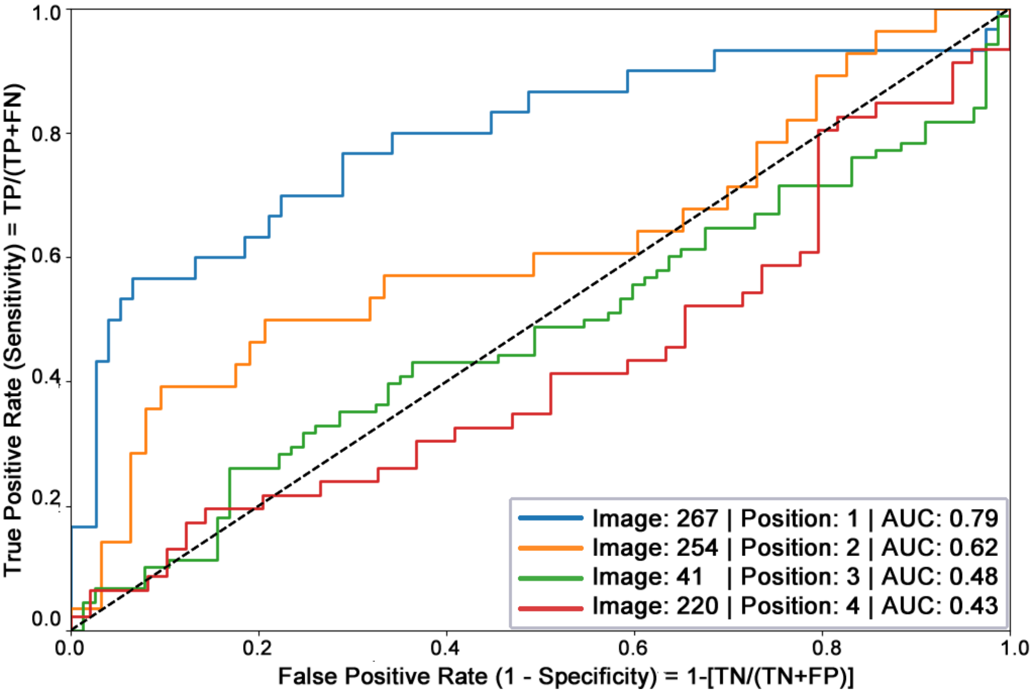


Figure 9. The ROC Curves of the Best and Worst Image Analyzed by the Sklearn Classifiers.

ROC curves are above the dashed line, and Images 41 and 220 are near or below this line. Notably, the AUC values are higher for the best-classified images. Among the top-performing images, the highest AUC value is Image 267, with 0.79. On the other hand, Image 220, with an AUC value of 0.43, is also the lowest, considering the worst images.

4.2. Top-Performing Classifiers

Table 3 displays the performance of 34 classifiers of Sklearn on a dataset with 300 images. The classifiers were evaluated based on their ranking for each image, ranging from 1st to 5th place. The Total Appearances column shows how often each classifier appeared within the top 5 rankings. For example, the SVC appeared 129 times within the top 5 rankings. The columns 1st, 2nd, 3rd, 4th, and 5th indicate how many times each classifier ranked in each position. For instance, the SVC was ranked 1st 44 times, 2nd 23 times, and so on. The top-performing classifiers were the SVC, which appeared 129 times within the top 5 rankings and ranked 1st 44 times, followed by the KNeighborsClassifier, which appeared 111 times within the top 5 rankings and ranked 1st 34 times. The HistGradientBoostingClassifier ranked 1st 27 times and appeared 89 in the top 5 rankings. On the other hand, the CategoricalNB and RadiusNeighborsClassifier did not appear in the top 5 rankings at all, making them the worst-performing classifiers in this dataset.

Regarding the top-ranked SVC classifier in Table 3 and the SVM and NuSVC variants, the three algorithms used the same procedure to find the decision boundary that maximizes the separation between data classes realized through the support vector. They work appropriately with high-dimensional data and for binary and multi-class classification. However, SVM uses a non-linear kernel, while SVC implements SVM with a linear kernel. The kernel refers to mathematical algorithms needed to change the dimensional space of a specific data set. NuSVC is also based on SVM but uses a particular parameter (Nu) to adjust the number of support vectors required to construct the decision boundary [Chang and Lin 2011, Aggarwal et al. 2015, Bao et al. 2014]. For comprehensive information on all 34 classifiers, please refer to the documentation at `scikit-learn.org`.

5. Conclusions and Future Works

The *LookASD* is a low-cost system, open source for the community to test and improve the method presented in this paper. By identifying visual attention patterns that correlate with ASD, this system provides healthcare professionals with an effective tool to assist in diagnosing children with ASD.

As the objective of this work was not to test the *LookASD* system with human beings, 80% of the dataset was used for training and 20% for testing. During training, 34 machine-learning classifiers from Sklearn were applied to the system, and the Accuracy, Sensibility, Specificity, Precision, RMSE, MAR and F1-Score metrics were calculated. After analyzing the results, the most accurate classifiers were SVC (a variation of Support Vector Machine – SVM), K-Nearest Neighbors and Histogram Gradient Boosting.

The results of this study suggest several potential future directions to improve the overall performance and usability of the *LookASD* system. One possibility is to use eye-trackers instead of webcams to enhance image accuracy. Furthermore, utilizing other datasets, when available, can improve the predictions. The usability test is needed to improve the *LookASD* interface and the reports generated for specialists. For this, the ethics committee of the UFABC has already approved the test to be performed by adults without ASD (Certificate of Presentation for Ethical Consideration – CPEC n: 60989422.0.0000.5594). In addition, in the current version of the *LookASD* system, the images the child views on the computer screen are chosen randomly. However, the prediction improves if the system selects the best-ranked images, and in the classifiers, alternative parameters beyond the defaults should be investigated. Finally, the *LookASD* system must be validated in the real world by children with and without ASD.

Data Availability Statement

The data that support the findings of this study are available at `https://github.com/jarriv/LookASD`.

Table 3. Top-Performing Classifiers.

Name	Total Appearances					
	1ª	**2ª**	**3ª**	**4ª**	**5ª**	**Total**
SVC	44	23	29	21	12	129
KNeighborsClassifier	34	17	16	23	21	111
HistGradientBoostingClassifier	27	16	19	18	9	89
RandomForestClassifier	21	14	18	14	21	88
QuadraticDiscriminantAnalysis	18	19	19	13	12	81
ExtraTreesClassifier	16	19	25	9	12	81
GaussianNB	10	22	18	17	10	77
GradientBoostingClassifier	8	21	13	13	17	72
BaggingClassifier	10	22	6	16	17	71
NuSVC	26	22	6	7	7	68
LinearDiscriminantAnalysis	2	7	14	18	25	66
AdaBoostClassifier	15	12	16	12	11	66
LogisticRegression	5	5	9	16	18	53
BernoulliNB	13	13	14	6	6	52
LogisticRegressionCV	5	9	7	11	16	48
DummyClassifier	0	8	11	20	6	45
CalibratedClassifierCV	4	10	12	8	7	41
GaussianProcessClassifier	3	9	6	5	8	31
NearestCentroid	5	5	7	6	7	30
RidgeClassifier	0	0	5	6	16	27
DecisionTreeClassifier	4	3	4	8	7	26
ComplementNB	6	2	6	5	4	23
MultinomialNB	0	5	6	5	7	23
ExtraTreeClassifier	1	6	3	5	4	19
LabelPropagation	5	3	3	2	5	18
MLPClassifier	8	0	1	5	3	17
LabelSpreading	0	5	3	3	2	13
RidgeClassifierCV	0	0	0	5	6	11
PassiveAggressiveClassifier	4	0	2	0	1	7
Perceptron	2	2	1	0	1	6
SGDClassifier	2	0	1	2	1	6
LinearSVC	2	1	0	1	1	5
CategoricalNB	0	0	0	0	0	0
RadiusNeighborsClassifier	0	0	0	0	0	0
Total	300	300	300	300	300	1500

References

Aggarwal, A., Sahay, T., and et al. (2015). Grid search analysis of nu-SVC for text-dependent speaker-identification. In *INDICON-IEEE*, pages 1–5.

Bao, Y., Wang, T., and Qiu, G. (2014). Research on applicability of SVM kernel functions used in binary classification. In *CSAIT-Springer*, pages 833–844.

Biasão, M. d. C. R. (2019). *Classificação da gravidade do transtorno do espectro autista baseada no padrão de rastreamento do olhar*. PhD thesis, USP.

Bortoletti, H. S. (2022). *Aquisição de vídeos e processamento de imagens faciais em dispositivos móveis para auxílio ao diagnóstico do Transtorno do Espectro Autista*. PhD thesis, USP.

Chang, C.-C. and Lin, C.-J. (2011). LIBSVM: a library for support vector machines. *ACM transactions on intelligent systems and technology (TIST)*, 2(3):1–27.

Cournapeau, D. (2023). Sklearn classifiers.

Donvan, J. and Zucker, C. (2017). *Outra sintonia: a história do autismo*. Editora Companhia das Letras.

Duan, H., Zhai, G., and et al. (2019). A dataset of eye movements for the children with autism spectrum disorder. In *Proc. of the 10th ACM MMSys*, pages 255–260.

Gribskov, M. and Robinson, N. L. (1996). Use of receiver operating characteristic (roc) analysis to evaluate sequence matching. *Computers & chemistry*, 20(1):25–33.

Han, J. and Kamber, M. (2006). Data mining: Concepts and techniques second edition. morgan kauffman. *San Fransisco*.

Klin, A. (2006). Autismo e síndrome de asperger: uma visão geral. *Brazilian Journal of Psychiatry*, 28:s3–s11.

Larose, D. T. (2015). *Data mining and predictive analytics*. John Wiley & Sons.

Nogueira Filho, A. d. L. (2020). Desenvolvimento de uma ferramenta computacional para o auxílio na análise do processamento holístico facial do transtorno do espectro do autismo (tea): estudos não clínicos. Master's thesis, UnB.

Ojala, M. and Garriga, G. C. (2010). Permutation tests for studying classifier performance. *Journal of machine learning research*, 11(6).

Oliveira, G. (2009). Autismo: diagnóstico e orientação parte i-vigilância, rastreio e orientação nos cuidados primários de saúde. *Acta pediát. port.*, 40(6):278–287.

Pal, M. and Rubini, P. (2022). Multimodal autism detection in children through facial and speech emotion. In *Rising Threats in Expert Appl. and Solut.*, pages 683–692.

Papoutsaki, A., Sangkloy, P., and et al. (2016). Webgazer: Scalable webcam eye tracking using user interactions. In *Proc. of the 25th Intern. Joint Conf. on Artificial Intelligence (IJCAI)*, pages 3839–3845. AAAI.

Quilici-Gonzalez, J. A. and Zampirolli, F. A. (2014). *Sistemas inteligentes e mineração de dados*. Triunfal Gráfica e Editora.

Simmons, D. R., Robertson, A. E., and et al. (2009). Vision in autism spectrum disorders. *Vision research*, 49(22):2705–2739.

Usta, M. B., Karabekiroglu, K., and et al. (2019). Use of machine learning methods in prediction of short-term outcome in autism spectrum disorders. *Psychiatry and Clinical Psychopharmacology*, 29(3):320–325.

Vapnik, V. (1999). *The nature of statistical learning theory*. Springer science & business media.