

Applications of Machine Learning Methods to Assist the Diagnosis of Autism Spectrum Disorder

Mahmoud Elbattah^{1,2}, Romuald Carette³, Federica Cilia⁴,

Jean-Luc Guérin¹ and Gilles Dequen¹

¹MIS Lab, University of Picardie Jules Verne, Amiens, France ²Faculty of Environment and Technology, University of the West of England, Bristol, United Kingdom ³Evolucare Technologies, Villers-Bretonneux, France ⁴CRP-CPO Lab, University of Picardie Jules Verne, Amiens, France

5.1 Introduction

Autism spectrum disorder (ASD) is described as a pervasive developmental disorder, which is characterized by a triad of social impairments including communication problems, difficulties with reciprocal social interactions, and repetitive behavioral patterns [1]. ASD-diagnosed individuals usually suffer from troubles in interaction and communication in multiple forms. The most remarkable symptom is the poor development of nonverbal skills such as the lack or absence of eye contact. With such troubling deficits, a considerable strain can unfortunately be placed on the well-being of autistic individuals and their families as well. From an economic standpoint, it was estimated that autism costs the UK, for

example, more than heart disease, cancer, and stroke combined [2].

The detection of autism at an early stage of development is highly favorable to realize common benefits for children and their families. However, the diagnosis of autism has been considered as a challenging task. Typically, the diagnosis process includes a variety of cognitive tests that could require hours of intensive clinical examinations. Furthermore, the lack of a gold standard test and the diversity of symptoms increase the complexity of identifying an accurate classification. In this regard, a range of technologies have been embraced for providing assistance over the course of autism diagnosis. Equally important, the integration with contemporary artificial intelligence (AI) approaches is currently allowing for further

potentials to advance the diagnostic applications of autism.

This chapter presents and discusses applications of integrating machine learning (ML) and eye-tracking to support the diagnosis of ASD. In general, the study adopts a vision-based approach using the visual representation of eye-tracking scanpaths as a means for learning the behavioral gaze patterns of autism. The core idea is to render gaze movements into a compact image format, which also maintains the dynamic characteristics of motion (e.g., velocity) using gradients of color. Accordingly, the detection of autism could be approached as a typical problem of image classification. This study has been initiated in the frame of an interdisciplinary research of psychology and AI. The collaboration is aimed to bring together psychologists from the CRP-CPO lab along with ML researchers from the MIS lab, based at the University of Picardie Jules Verne in France.

This chapter is structured into two parts as follows. First, a review is provided on the literature related to supporting the ASD diagnosis using ML techniques. The literature review is aimed to provide the state of the art of diagnostic tools in the autism context. Second, the chapter goes through a practical aspect that included the development of a variety of ML models. On one hand, unsupervised ML is applied to explore potential data-driven insights within the image dataset of eye-tracking scanpaths. On the other hand, supervised learning included the development of a classification model to detect autism. Broadly, the experimental results demonstrate the potentials of ML for supporting the diagnosis of ASD. The chapter is generally aimed to be a continuation and elaboration of our earlier work [3,4].

5.2 Background and related work

This section aims to review the ASD literature from two aspects of relevance. Initially, the first part reviews the studies that employed

eye-tracking methods to analyze the gaze behavior among ASD-diagnosed individuals. Further, a brief history is given on the origins of eye-tracking. Subsequently, we provide representative contributions that attempted to combine eye-tracking with ML approaches to support the diagnosis process of autism. The review is selective rather than exhaustive in the sense that it highlights various approaches of applying ML in this context.

5.2.1 Analysis of visual attention in autism

Making an accurate diagnosis of autism has been considered as a challenging task, which usually requires an intensive clinical assessment and experience. The early efforts aimed at developing diagnostic instruments including observational measures or diagnostic interviews such as Childhood Autism Rating Scale (CARS) [5], Autism Diagnostic Interview [6], and Autism Diagnostic Observation Schedule (ADOS) [7]. With contemporary advances in technology, new approaches have come into prominence to assist the procedures of diagnosis and assessment. Examples include a variety of technologies such as electroencephalography [8], Magnetic Resonance Imaging [9], and eye-tracking in particular.

Eye-tracking refers to the process of capturing and measuring eye movements and the absolute point of gaze (POG) [10]. The POG represents the focal point of the eye gaze in the visual scene. Interestingly, the use of eye-tracking has a quite long history that dates back to the 19th century. The early development of eye-tracking is credited to the French ophthalmologist *Louis Javal* from the Sorbonne University. In his seminal research that commenced in 1878, Javal had produced the novel observations of fixations and saccades based on the gaze behavior during the reading process [11,12]. A fixation describes the brief moments while the eye gaze is paused on a

particular object, which allow the brain to perform the perception process. The average duration of fixation was estimated to be around 330 Ms [13]. While saccades include a constant scanning with very rapid and short eye movements. Saccades consist of quick ballistic jumps of 2° or longer, which continue for about 30–120 Ms [14].

Afterwards in 1908, *Edmund Huey* built a primitive eye-tracking tool for analyzing eye movements while reading [15]. Further advanced implementations of eye-tracking instruments were developed by *Buswell* [16,17]. Photographic films were utilized to record the eye movements while looking at a collection of paintings. The eye-tracking records included both of the direction and duration of movements. While modern eye-trackers largely fall into three categories as: (1) Screen-based eye-trackers, (2) Eye-tracking glasses, and (3) virtual reality headsets. Eye-trackers have been utilized in a multitude of applications. To name a few, applications of marketing [18], psychology [19], product design [20], and many others.

In the autism context, the eye-tracking technology has particularly received wide interest whereas abnormalities of the eye gaze are largely recognized as the hallmark of ASD. As such, the literature contains abundant contributions that endeavored to apply eye-tracking methods for analyzing and understanding the behavioral characteristics of autism. Various interesting physiological elements were reported based on findings output from eye-tracking experiments. For instance, eye movements in face-to-face interactions were observed to be different for individuals who had different levels of the autism severity [21]. Specifically, highly autistic individuals were noticed to experience saccades of shorter duration and less frequency as well. In another application of eye-tracking, ASD toddlers could be identified based on the frequency of saccades and fixations [22]. The results showed that the ASD-diagnosed group spent significant longer periods of fixations on dynamic geometric images.

Another longitudinal study inspected the patterns of fixations for a group of infants in the range of 2–6 months [23]. They notably indicated that the infants who eventually developed ASD exhibited a mean decline in fixations as opposed to the typical group. More recent studies (e.g., [24,25]) proposed eye-tracking-based methods to establish objective measures that could quantify the risk of developing autism or estimating the severity of symptoms.

5.2.2 Machine learning for autism diagnosis

The ML concept was originally popularized by *Arthur Samuel* in 1959. Samuel described ML as the subfield of Computer Science that gives computers the ability to learn without being explicitly programmed [26]. In contrast to traditional programming, ML seeks to extrapolate algorithms from data exclusively. As such, the power of ML is that it allows for extracting insights, making predictions, or taking actions with minimal human intervention (if any). That approach of “automatic” learning has become very attractive to a diversity of domains including recommender systems [27,28], marketing personalization [29,30], medical applications [31,32], and many others. Figs. 5.1 and 5.2 highlight the distinction between traditional programming and ML from a computational perspective.

The development of ML models can be broadly organized into supervised or unsupervised learning. On one hand, supervised ML deals with labeled samples of data, where the desired output is known precisely. The learning algorithm receives a set of inputs along with corresponding labels, and the model can learn by comparing its predictions against the ground truth. Through an optimization process, the error between the actual and desired outputs can be iteratively minimized. On the other hand, unsupervised ML uses training data that do not include any output information (i.e. labels).

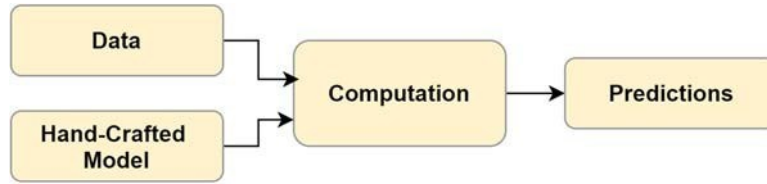


FIGURE 5.1 Traditional programming.

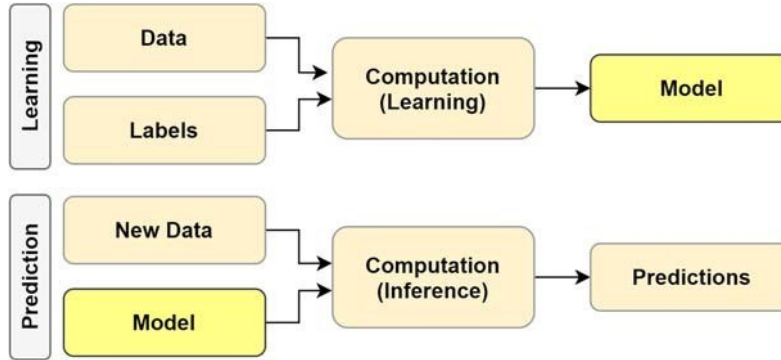


FIGURE 5.2 Machine Learning.

Unsupervised models (e.g., clustering, association rules) can provide descriptive knowledge to help understand the inherent structure or properties of data. For instance, the Apriori algorithm [33,34] seeks to find out frequent itemsets in a dataset of transactions.

Beyond that, self-supervised learning is an emerging approach that is rapidly becoming the central focus of ML research (e.g., [35,36]). Self-supervised learning can be regarded as a form of unsupervised learning, however the data itself are used to provide the supervision. The approach includes the development of (pretext) tasks formulated based on unlabeled data. The pretext tasks are used for learning representations, which can be employed for tackling further tasks of interest.

A variety of ML techniques have been used in tandem with eye-tracking tools to leverage further potentials. Eye-tracking devices typically produce abundant amounts of eye-gaze information. A few minutes of operating time

would output thousands of records regarding the gaze position and eye movements. With large amounts of data, ML has been considered as an ideal path in a variety of applications.

In this respect, the literature includes a variety of ML applications in different modalities and for different purposes. For instance, a study aimed to analyze the eye focus on faces in conversations for Fragile X Syndrome children [37]. A set of classification models were developed including Support Vector Machine, Naive Bayes, Recurrent Neural Network (RNN), and Hidden Markov Model. The RNN model could realize a high classification accuracy, as 86% and 91% for female and male participants respectively.

Likewise, a promising accuracy could be achieved ($\approx 83\%$) to detect autism in a group of children aged 8–10 years [38]. Using a Long Short-Term Memory model, the classifier was trained using a set of features based on the dynamics of eye movement (e.g., acceleration

amplitude, duration). In another interesting application, eye-tracking data were collected based on different kinds of tasks related to browsing and searching web pages [39]. The dataset collected was used to train a classification model to help detect ASD-diagnosed participants.

Another study employed tablets equipped with touch-sensitive screens and sensors for recording the kinematics of movement [40]. A set of 262 features were extracted, which computationally described the movements sensed by the device. The subjects included 37 ASD-diagnosed children, and another 45 typically developing (TD). A set of ML models were experimented using Extra-Trees [41], Random Forest [42], and Regularized Greedy Forest [43]. The movement patterns could identify ASD participants with accuracy up to 93%. In another eye-tracking application [44], a facial emotion recognition task was used to detect autism. Eye-tracking data were collected from ASD and TD individuals under the Dynamic Affect Recognition Evaluation task [45]. A random forest model was trained for the classification task, which could achieve over 75% accuracy of prediction.

In addition, the recent advances in Deep Learning [46] are opening new frontiers for supporting the diagnosis process of autism. Deep neural networks (DNNs) are composed of multiple processing layers, which allow for learning hierarchical representations with multiple levels of abstraction of data. Architectures of Convolutional Neural Networks (ConvNets) [47,48] and RNNs [49] have been successfully implemented for complex tasks such as Machine Translation and Computer Vision (e.g., [50,51]).

A variety of DNN implementations have been employed for building classification models for detecting ASD. For instance, deep learning was used in a framework for the automatic screening of ASD [52]. It was proposed to incorporate two different modalities related to the human visual attention including attentional preference captured from photo-taking and image-viewing

tasks. The temporal information in eye-tracking data was utilized, which suggested behavioral differences between the ASD-diagnosed and TD individuals. In another application, a DNN-based model was developed to detect ASD-diagnosed individuals based on eye-tracking data in free-image viewing [53]. Deep Learning was employed to automatically extract features from a selection of discriminative images.

Further studies experimented the use of visual representations of eye-tracking records. As such, predictive modeling could be implemented from the perspective of image analysis or classification. In this regard, a recent work [3] reported an excellent accuracy (ROC-AUC \approx 0.90), though using simple architectures of Neural Network models. The eye-tracking scanpaths were transformed into images, which were used to train the classifiers. The key idea was to produce visualizations of the eye-tracking scanpaths, which could depict the geometric patterns of gaze and its dynamics as well. As an application of unsupervised learning, another study [4] used scanpath images as a means for data clustering.

5.3 Data description

5.3.1 Participants

The participants included a set of 59 children recruited from a number of schools in the region of Hauts-de-France. The age of participants ranged from 3 to 12 years old. It was highly desirable for participants to be at an early stage of the ASD development. A parental permission was obtained for every child before taking part in our experiments. Further, the parents were acquainted with the study objectives through orientation sessions.

Initially, the participants were organized based on a basic binary grouping as (1) TD and (2) ASD-diagnosed. In addition, the CARS score [5] was employed to classify the severity of autism more precisely. The CARS method

TABLE 5.1 Summary of study participants.

Count of participants (TD, ASD)	59 (30, 29)
Gender (female, male)	21 (= 36%), 38 (= 64%)
Age (mean, median)	7.88, 8.1 years
CARS score (mean, median)	32.97, 34.50

has been widely applied in the Psychology practice for describing the severity of ASD symptoms [54]. The scale includes various ratings on different behavioral aspects (e.g., verbal communication, activity level). Table 5.1 summarizes the characteristics of participants.

5.3.2 Experimental protocol

The eye-tracking experiments included a set of video scenarios, which were particularly designed to stimulate the eye gaze. The participants were seated at approximately 60-cm distance away from the monitor. A quiet room at the university campus was used for running our experiments. In addition, physical barriers were applied to avoid visual distractions.

We used a SMI Red-M eye tracker with 60 Hz sampling rate, which is a screen-based eye-tracker. The eye-tracker was operated along with a 17-inch monitor in our experiments. The screen resolution was 1280x1024.

The content and length of videos varied to allow for analyzing the ocular activity from different aspects and levels. The content of videos was generally aimed to include visual items (e.g., colorful balloons, cartoons), which could be attractive to children in particular. The position of items can change over the experiment timeline. In addition, parts of the scenarios included human presenters. The presenter would attempt to turn the participant's attention to elements, which could be visible or invisible around the display area. The French language was the working language in all videos and conversations, as the mother tongue

of participants. Fig. 5.3 presents a sample screenshot captured from one of the videos.

We used other stimuli provided by the SMI Experiment Center Software. The Stimuli included a variety of types, which are used in the eye-tracking research. For instance, static and dynamic naturalistic scenes with and without receptive language, static face or objects and cartoons stimuli, and other joint attention stimuli. Eye-tracking experiments usually took about 5 minutes. The participants were inspected with respect to the quality of eye contact with the presenter, and the level of focus on other elements. A five-point scheme of calibration was applied. A set of verification procedures followed the calibration scheme.

The eye-tracking device captured three categories of eye movements including fixations, saccades, and blinks. Additionally, the POG coordinates were captured by the eye tracker. The coordinates were specifically significant to implement our approach in terms of visualizing the gaze scanpath, and computing its dynamics (e.g., velocity). The implementation of this part is elaborated in the following section. Table 5.2 gives a simplified view of the eye-tracking records. As an example, the table lists five eye-tracking records that represent a sequence of two fixations and three saccades. The records also give the POG coordinates for the right and left eye over time. Due to limited space, many variables had to be excluded from the table below (e.g., pupil position, pupil diameter, pupil size).

5.3.3 Visualization of eye-tracking scanpaths

A scanpath represents a sequence of consecutive fixations and saccades as a trace through time and space that may overlap itself [55]. The term was first brought into use by *Noton* and *Stark* back to 1971 in a couple of publications [56,57]. Scanpaths are commonly utilized in the eye-tracking context as a practical means to depict the gaze behavior in a visual manner.



FIGURE 5.3 Sample screenshot.

TABLE 5.2 An example of the eye-tracking records.

Timestamp [Ms]	Eye movement category	Point—regard X [px]	Point—regard Y [px]	Pupil diameter—right [mm]	Pupil diameter—left [mm]
8005654.069	Fixation	1033.9115	834.0902	4.3785	4.5431
8005673.953	Fixation	1030.3754	826.0894	4.4050	4.5283
8005693.85	Saccade	1027.337	826.3127	4.4273	4.6036
8005713.7	Saccade	1015.0085	849.2188	4.3514	4.5827
8005733.589	Saccade	613.7673	418.1735	4.3538	4.5399

Fig. 5.4 represents a basic scanpath example, which includes a few fixations and saccades. As it appears, the fixations are shown as circles, and the saccades as lines connecting those fixations. The diameter of a fixation indicates the duration, while the lengths of lines represent the continuation of saccades.

The basis of our approach is centered on learning the visual patterns underlying eye-tracking scanpaths. Specifically, scanpaths are used as a means to compactly describe the gaze behavior into a visual representation that could simplify the ML process. Further, it was aimed to visually encode the dynamics of eye motion using color gradients. To this end, we made use of the POG coordinates in the eye-tracking records. Based on the POG displacement over time, the velocity of eye gaze was calculated. Subsequently, the eye

movement and its dynamics were transformed into a visual-based representation. For each participant, a set of scanpath images were constructed using the following procedures:

- A line is drawn for each transition from (x_t, y_t) to (x_{t+1}, y_{t+1}) , where t represents a point of time in the experiment.
- The change in color across lines was used to visualize the movement dynamics. Using the grayscale spectrum, the color values were tuned based on the magnitude of velocity (i.e., speed) with respect to time.
- Eventually, the images were vertically flipped whereas the origin was positioned at the bottom of the screen.

A threshold was applied to limit the number of visual points included in images. The threshold

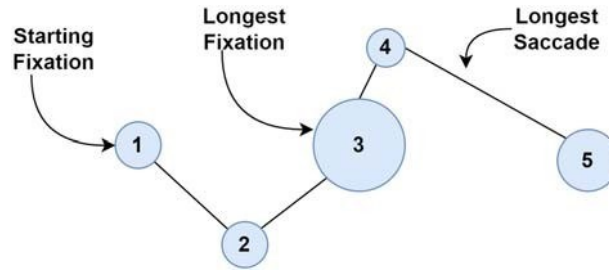


FIGURE 5.4 A typical example of an eye-tracking scanpaths [55].

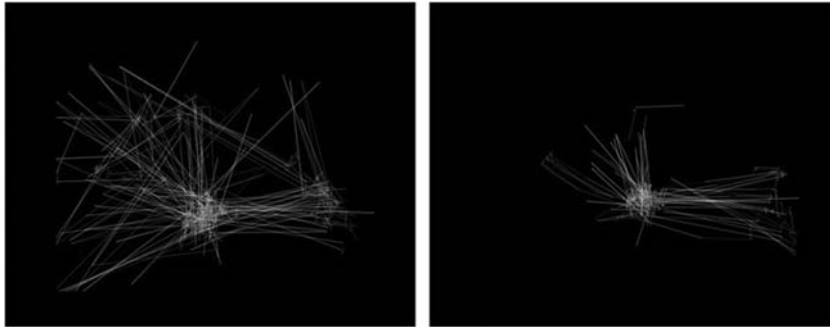


FIGURE 5.5 Visualization of eye-tracking scanpaths. The left-sided image represents an ASD-diagnosed participant, while the other is for non-ASD.

was aimed to be at a sufficient level for describing the gaze behavior patterns. Several tests were performed for choosing an appropriate value in this regard. Eventually, it was decided to set the threshold 5 200, which turned out to capture the key features of movement, and helped avoid cluttered visualizations as well. The visualizations were produced using the Matplotlib library [58].

The outcome of the visualization process was an image dataset containing more than 500 samples. Specifically, 328 images related to the TD participants, and another 219 images for the ASD-diagnosed. The default image dimensions were set as 640x480. The dataset along with its metadata files have been made publicly available on the *Figshare* repository [59]. It is hoped for the dataset to be beneficial for further studies or applications in the autism context. Fig. 5.5 presents a couple of scanpath visualizations related to ASD and TD participants.

5.4 Unsupervised learning: clustering of eye-tracking scanpaths

In this part, we apply unsupervised learning using image clustering as a means to analyze the characteristics of the gaze behavior involved in autism. We attempt to apply data clustering to discover potential clusters within the image dataset. The key idea is to learn data-driven clusters based on the visual representation of eye-tracking scanpaths. We were generally interested in discovering patterns, structures, or associations underlying the scanpath visualizations produced by our eye-tracking experiments. The study was specifically motivated by the questions below:

1. Would the scanpath visualizations indicate an inherent clustering structure?
2. If so, could the cluster analysis reveal potential connections pertaining to the

dynamics of gaze behavior, such as the velocity or acceleration of movement?

3. Furthermore, could the clusters demonstrate interesting patterns with regard to the characteristics of participants, such as age for example?

The clustering model was developed based on features learned by a deep autoencoder. The following sections elaborate the process of feature extraction from images.

5.4.1 Image preprocessing

A set of image-processing procedures were implemented to help reduce the problem dimensionality. Initially, the image dimensions were consistently scaled down to 100x100. For further simplification, the images were transformed into a grayscale format. The grayscale transformation reduces the visual representation by eliminating the hue and saturation elements while retaining the luminance. As such, the set of features was significantly reduced from 30 K (i.e., 100 3 100 3 3) into 10K (i.e., 100 3 100 3 1).

5.4.2 Feature extraction using principal component analysis and t-SNE

The initial set of features represented the grayscale values, which consisted of 10 K features altogether. Unavoidably, the raw representation included a large redundant proportion, which was part of the blank background of images. That part of images could be largely regarded as noise that should be discarded. The sensitivity of clustering algorithms (e.g., K-Means) to noise was particularly addressed in the ML literature (e.g., [60,61]).

In this respect, we applied techniques of data compression to compactly reduce the raw features set and mitigate the impact of background noise. Initially, the principal component analysis (PCA) was used to transform the images into a compressed format. Specifically, the 10K features

were reduced into a much smaller set of 50 components only. Further, the t-Distributed Stochastic Neighbor Embedding (t-SNE) technique [62] was utilized for reducing the dataset dimensionality. The t-SNE method is used for performing a non-linear dimensionality reduction, which can map data of high dimensionality into a lower dimensional space of two or three dimensions.

5.4.3 Feature extraction using deep autoencoder

Autoencoders are regarded as a special implementation of Artificial Neural Networks (ANNs). In an unsupervised manner, autoencoders seek for learning compressed representations of data, referred as “codings.” In contrast to typical ANN applications (e.g., regression, classification), autoencoders are developed in a completely unsupervised manner. As such, training an autoencoder does not need any label information. The functions of compression and decompression are automatically learned from data samples rather than hand-crafted. Fig. 5.6 illustrates the basic architecture of autoencoders, with an example of an autoencoder used to re-construct an image.

The idea of autoencoders was originally introduced in the 1980s by the PDP (Parallel Distributed Processing) group including Geoffrey Hinton, at the University of California San Diego. They were motivated by the challenge of training a multilayered ANN, which can allow for learning any arbitrary mapping of input to output [64]. That work led to the backpropagation algorithm, which has become the standard approach for training ANNs.

There is a diversity of valid applications that could be realized by autoencoders. Fundamentally, autoencoders can be effectively used as a means of dimensionality reduction (e.g., [65,66]), whereas codings represent a latent space of significantly lower dimensionality compared to the original input. Furthermore, they can perform the functionality of generative modeling. The

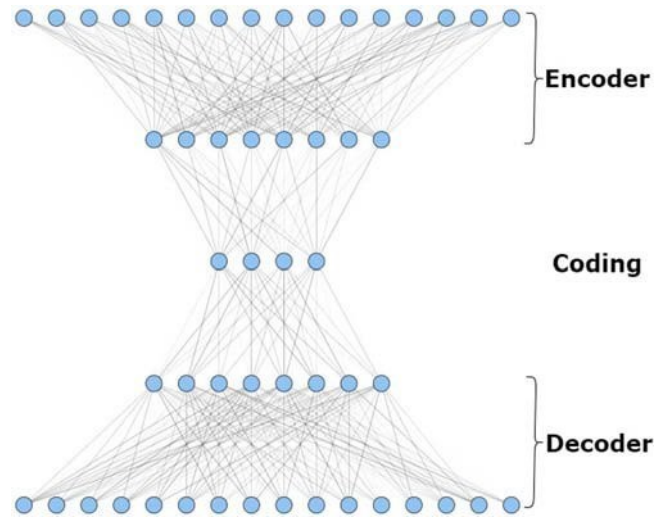


FIGURE 5.6 The architecture of autoencoders [63].

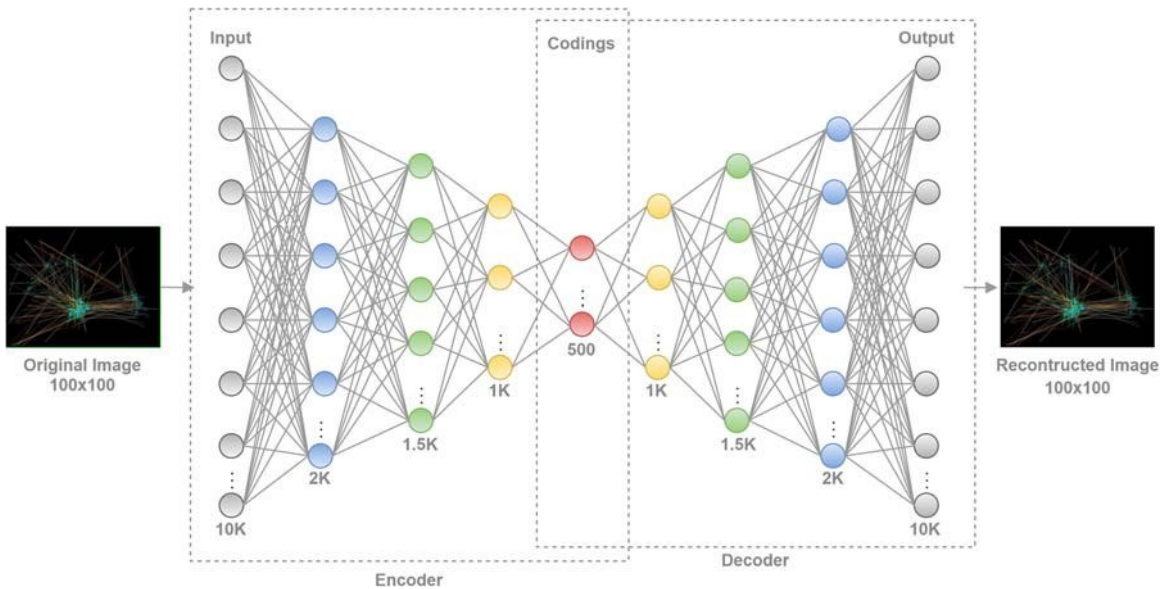


FIGURE 5.7 The autoencoder architecture.

codings learned can be utilized to randomly generate synthetic samples, which are similar to the original data. More interestingly, autoencoders provide a potent mechanism for feature extraction, which is the case in the present work.

Fig. 5.7 sketches the autoencoder architecture used in our experiments. As it appears, a multi-layered ANN performed the functionality of encoding and decoding. On one hand, the encoder consists of five layers that take an input

matching the image dimensions. The compression process is conducted through a sequence of four hidden layers. The concentration of neurons continues to gradually decrease down to 500 units, which contain the compressed codings. In this manner, the image codings represented only %5 of the original dimensionality.

On the other hand, the decoder can be basically regarded as a flipped duplicate of the encoder. In an inverse fashion, the number of neurons progressively increases all the way back to the original dimension (i.e. 10k units). The autoencoder could be trained by comparing the decoder output against the original images processed by the encoder. The Rectified Linear Unit (ReLU) was used as the activation function in all layers. The training process was completed over 100 epochs using the Adam optimizer [67] with its default parameters. The dropout technique [68], with 40%, was applied to minimize the possibility of overfitting. Fig. 5.8 plots the model loss in the train and validation sets over 100 epochs for the autoencoder training.

Fig. 5.9 demonstrates the autoencoder output against the original input of images. As shown, the figure compares two image samples against the output reconstructed by the autoencoder. Despite the output images were

not identical, it appears that they could largely capture the main features of the original samples. The autoencoder implementation was accomplished using the Keras framework [69] along with the TensorFlow backend [70].

5.4.4 K-Means clustering

The K-Means algorithm was used for developing the clustering model. With its conceptual simplicity, the K-Means approach has been successfully implemented in clustering-related tasks across a very wide range of application domains. The basic idea of K-Means is to find a clustering structure that minimizes some distance metric (e.g., Euclidean distance) as a measure of similarity among data points and their representative values (i.e. centroids). The centroid is calculated as the mean of all instances in a cluster. The Sum of Squared Error is used to measure the total squared distance of points to their centroids as in the equation below.

$$J(C_K) = \sum_{x_i \in C_K} \|x_i - \mu_K\|^2$$

where μ_K is the mean of cluster C_K , and $J(C_K)$ is the squared error between μ_K and the points in C_K [71].

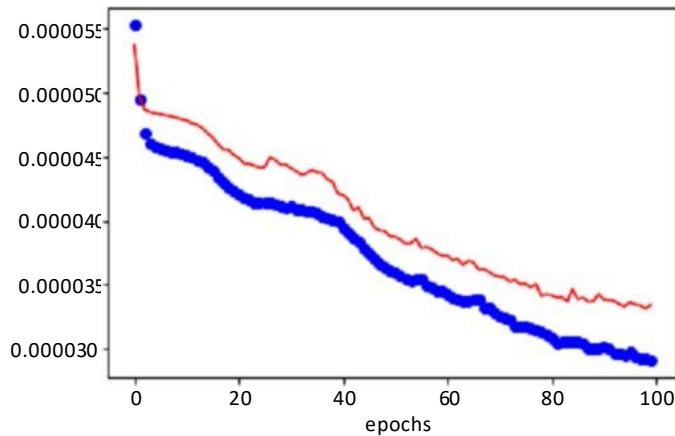


FIGURE 5.8 Training loss (below) and validation loss over 100 epochs.

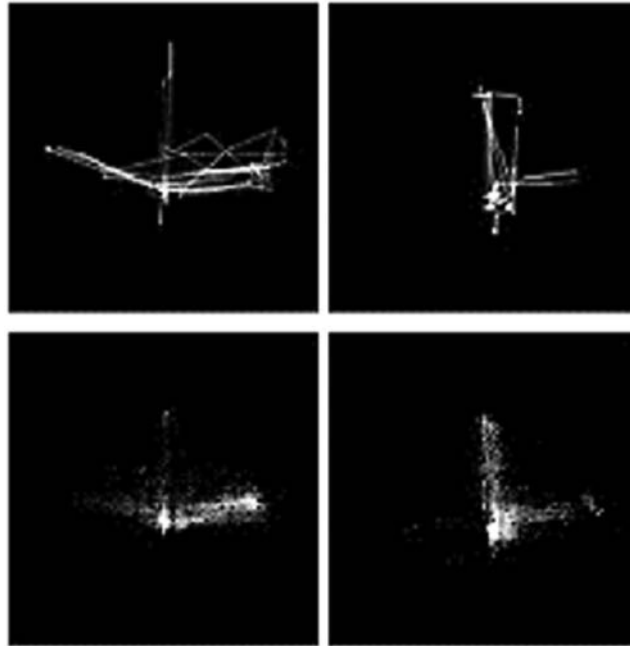


FIGURE 5.9 Sample images produced by the compressed codings. The upper images represent the original input, while images below represent the output reconstructed by the autoencoder.

Over a specified number of iterations, the algorithm seeks to partition the data space into a predefined number of clusters (i.e., K). The algorithm can converge to a solution in case that the cluster assignments are not changing anymore, or the number of iterations has been executed.

With the K-Means approach, the initial point of concern is usually about choosing the suitable number of clusters (K). In our case, there was already a rational assumption about the number of clusters that could conceivably exist in the dataset. The simplest grouping of data is a binary structure (i.e., $K = 2$), which would resemble the original grouping of participants (i.e., TD or ASD). While a fine-grained representation is to break down the ASD set into smaller groups in a bid to reflect the severity of autism. Based on the severity of ASD symptoms, the CARS scheme classifies the autism spectrum into three groups as low, mild, and severe (see Table 5.3). As such, the

TABLE 5.3 Grouping of autism spectrum disorder (ASD) based on Childhood Autism Rating Scale (CARS) [5].

Category of ASD	CARS score
Low	CARS ≤ 30
Mild	$30 < \text{CARS} \leq 36$
Severe	CARS ≥ 36

K-Means model was experimented using $K = 5$ ($2 < K < 4$). The Scikit-Learn library [72] was used to implement the clustering models.

5.4.5 Quality of clusters

The Silhouette method [73] was used to inspect the quality of clusters. The Silhouette score presents an objective means for measuring the robustness of cluster membership. As

such, it has been widely utilized in clustering-related tasks. The basic idea is to examine the distance of separation between the clusters by measuring the proximity of each point to other points located in neighboring clusters.

The Silhouette score represents a continuous range of $[-1, 1]$, where scores near 1 positively indicate that a point is far away from other clusters. In contrast, values closer to 0 can be interpreted as being located on, or very close to, the decision boundary with neighboring clusters. While negative scores strongly suggest that a point may have been assigned to the wrong cluster. The Silhouette score is calculated as below:

$$S(i) = \frac{(b(i) - a(i))}{\max(a(i), b(i))}$$

where $a(i)$ is the average distance of point i to all other points in its containing cluster and $b(i)$ is the smallest average distance of point i to points in another cluster (i.e., distance to the closest neighbor cluster).

Fig. 5.7 compares the Silhouette scores using a different set of features for training the clustering model. Apparently, the highest score was achieved when $K = 2$ in all cases. However, the quality of clusters declined after applying further partitioning of clusters (i.e. $K = 3, 4$). In general, there was a poor separation of clusters in case of

using the raw set of features, or other lower representations using PCA and t-SNE (Fig. 5.10).

In contrast, the autoencoder-learned features could notably improve the quality of clusters. For $K = 2$, the Silhouette score increased to more than 0.6. Likewise, the quality of clusters could largely be maintained for $K = 3, 4$. This obviously demonstrates the autoencoder capacity for extracting efficient features. In view of the clustering experiments, it was indicated that there was an underlying clustering structure based on the visual representation of eye-tracking scanpaths. This largely provides an answer to part of the questions addressed earlier.

5.4.6 Cluster analysis

An exploratory analysis is given in this section to provide an interpretation of the clusters and their characteristics. It was basically aimed to investigate possible correlations, which could be linked to the particular gaze behavior in autism. Specifically, the clusters were examined with respect to the velocity and acceleration of eye movement.

The general characteristics of clusters, where $K = 2$, are summarized in Table 5.4. The table compares the clusters in terms of the ASD proportion, and average age. It turned

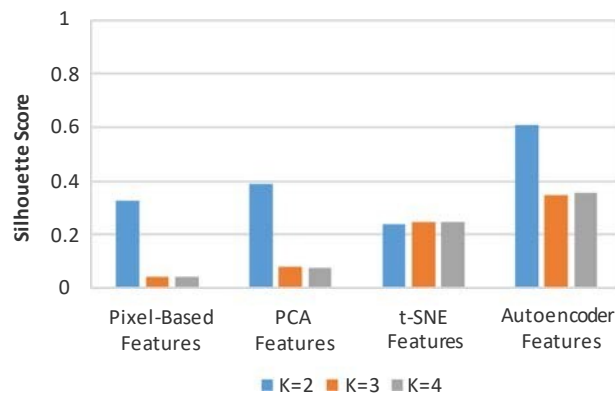


FIGURE 5.10 Quality of clusters based on the Silhouette score.

TABLE 5.4 Summary of clusters (k 5 2).

Cluster	ASD % (≈)	Avg. age (years)
Cluster 1	33%	8.2
Cluster 2	85%	8.7

TABLE 5.5 Summary of clusters (k 5 3).

Cluster	ASD % (≈)	Avg. age (years)
Cluster 1	28%	7.95
Cluster 2	49%	8.76
Cluster 3	94%	9.02

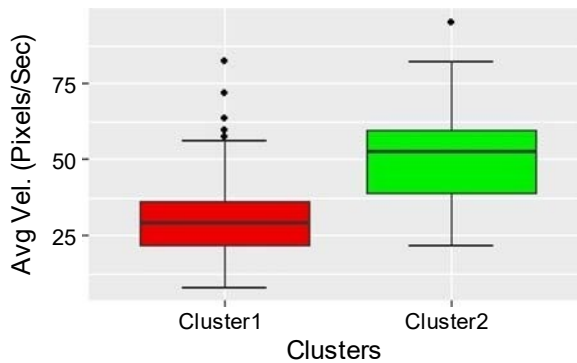


FIGURE 5.11 The variation of gaze velocity in clusters (K 5 2).

out that the percentage of ASD samples was notably significant in Cluster 2. This can indicate that the clusters included a coherent structure, which resembled the basic grouping of participants (i.e., ASD or TD).

On the other hand, Table 5.5 describes the clusters where K 5 3. It appears that the ASD proportion was much higher in Cluster 2 and Cluster 3. In general, the average age of participants was comparable across all clusters (K 5 2 or 3).

Fig. 5.11 plots the velocity of eye movement, for K 5 2. Interestingly, Cluster 2 was considerably higher in this regard, apart from a few

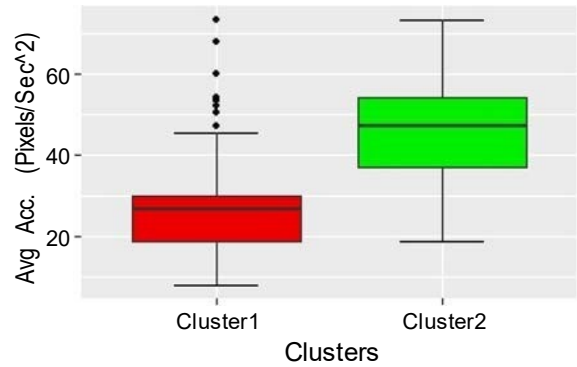


FIGURE 5.12 The variation of gaze acceleration in clusters (K 5 2).

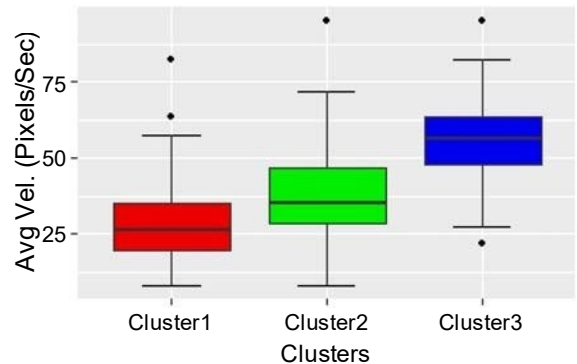


FIGURE 5.13 The variation of velocity in clusters (K 5 3).

outliers in Cluster 1. Similarly, Fig. 5.12 shows that the participants of Cluster 2 experienced a higher pace of gaze acceleration. It is noteworthy that the majority of Cluster 2 belonged to the ASD samples.

Similarly, Fig. 5.13 compares the clusters, where K 5 3, with respect to the velocity. It appears clearly again that the velocity of gaze was higher in clusters that included larger proportions of ASD-related samples. This could draw possible links related to the gaze behavior of ASD-diagnosed individuals and the dynamics of autistic gaze.

5.5 Supervised learning: classification model

This section includes the development of the classification model development, which can be used to detect autism. Using ML, our approach aims to learn the gaze patterns of ASD. The model was trained using the eye-tracking scanpath images as follows.

5.5.1 Data preprocessing and augmentation

A set of image-processing techniques was initially applied to help simplify the model training. First, the black background was cropped from images as much as possible. The cropping was implemented using the OpenCV library [74]. Second, all images were scaled down to dimensions of 256x256. Resizing the images could reduce the problem dimensionality by decreasing the number of features under consideration. The impact of resizing was also examined in the early ML experiments.

Further, augmentation techniques were applied to enlarge the image dataset, and increase the diversity of scanpath images. Augmentation is a prevalent approach to help models generalize better and reduce the risk of overfitting. The basic idea is to produce synthetic samples using a random set of image transformations such as rotation, shearing, or zooming for example. The application of augmentation was recognized to generally improve the prediction accuracy in image classification tasks (e.g., [75,76]). In our case, more than 2.5 K additional samples were synthetically produced. The application of data augmentation was largely facilitated, thanks to the Keras library [69], which comes with built-in functionalities for that purpose.

5.5.2 Model design

We designed a deep ConvNet model for the classification task. ConvNets typically include three categories of layers, including convolutional layers, pooling layers, and fully connected layers. The learning process goes through a series of convolutions and pooling, which break down the input image into a set of feature maps. Convolutional layers initially attempt to extract features from the image through applying a convolutional kernel all over the image. Subsequently, pooling layers work on reducing the dimensions of feature maps extracted.

Eventually, the output of this process usually feeds into a fully connected layer structure to produce the final prediction. In our case, the CNN model composed of four convolutional layers, four pooling layers, and two fully connected layers. In addition, dropout layers were utilized, which help reduce the possibility of overfitting.

5.5.3 Classification accuracy

The classification accuracy is analyzed based on the Receiver Operating Characteristics (ROC) curve. The ROC curve plots the relationship between the true positive rate and the false positive rate across a full range of possible thresholds. Fig. 5.14 plots the ROC curve of the ConvNet model. The figure also shows the approximate value of the area under the curve along with its standard deviation based on a threefold cross-validation. The model could notably achieve an excellent prediction accuracy (ROC-AUC \approx 0.9).

The model was implemented using the Keras [69] with TensorFlow backend [70]. Three rounds of cross-validation were applied to test the model accuracy. Training the model took \approx 3 minutes using a single Tesla K80 GPU. Fig. 5.15 demonstrates the model accuracy in train and validation sets over 20 epochs, while 20% of the training set was used for model validation.

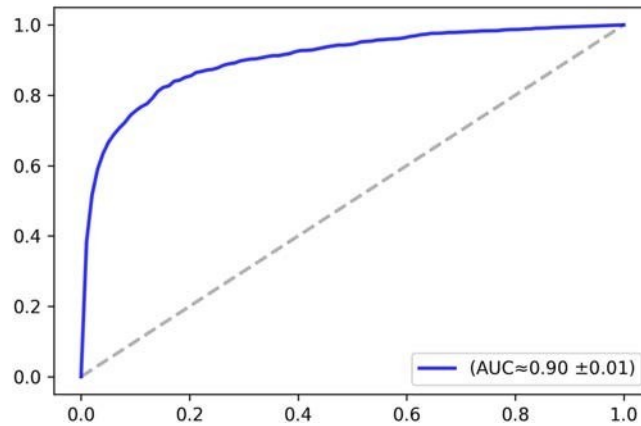


FIGURE 5.14 ROC curve.

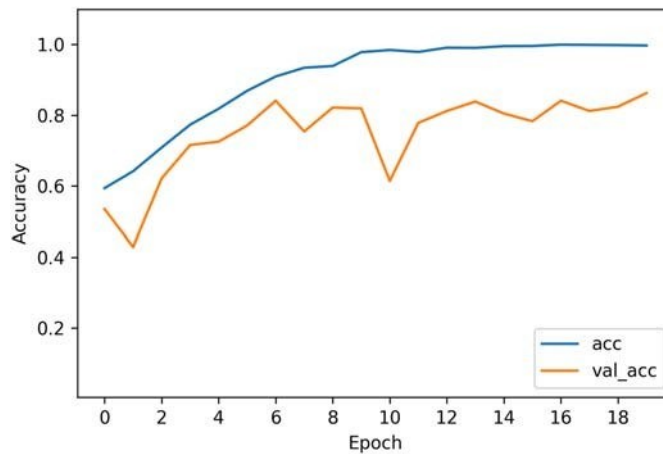


FIGURE 5.15 Model accuracy in training and validation.

5.6 Demo application

As a practical illustration, an application was developed to demonstrate our approach. The application links altogether the modules of eye-tracking, data visualization, and ML. The demo application consists of three main components including: (1) Presentation, (2) Web services, and (3) Prediction.

The presentation layer performs the main functionality of UI and application interactivity. The UI elements were implemented using

ASP.NET and the JQuery JavaScript library. The web services and prediction module were provided by the Azure ML platform. The Azure ML platform is employed to facilitate the model deployment as a web service. Specifically, Azure ML was used to host the classification model and the Python implementation used to visualize eye-tracking scanpaths. In this manner, the classification model could be accessed as a web service using request/response calls. Fig. 5.16 sketches the application architecture.

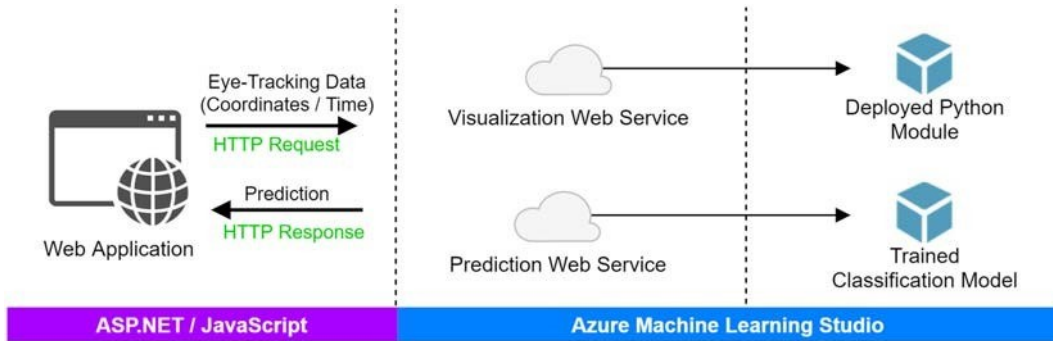


FIGURE 5.16 Application architecture.

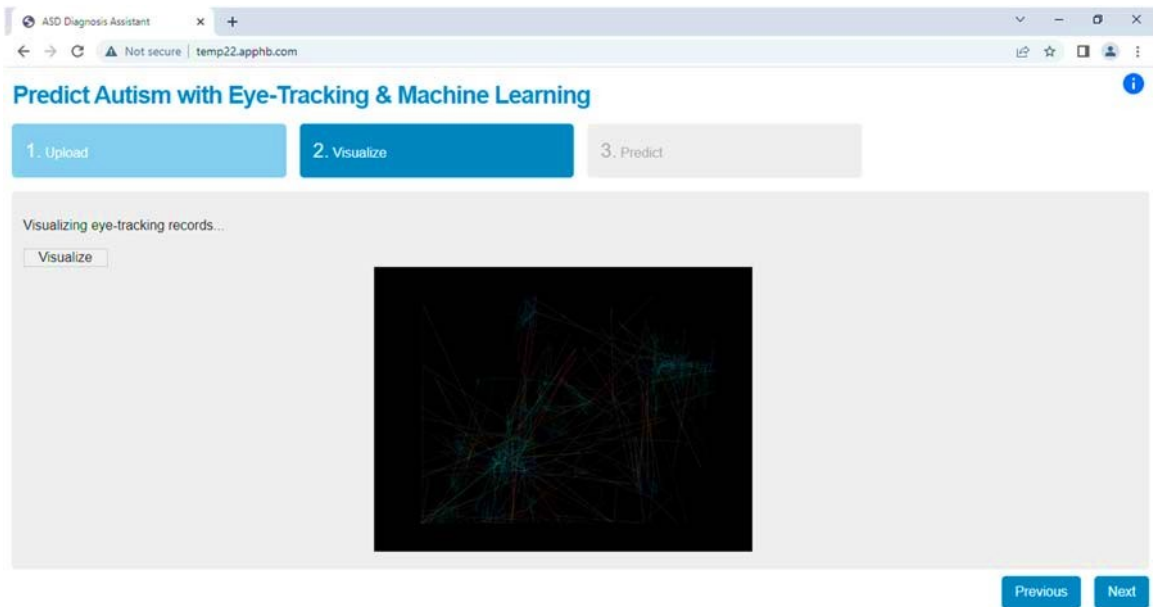


FIGURE 5.17 Screenshot of the demo application.

The application user is guided through the following procedures. First, the user should upload the eye-tracking dataset, which is expected to include the description of eye movements along with the POG coordinates. Second, the application produces the scan-path visualization. The visualization is rendered by calling a web service hosted on

Azure ML. Eventually, the application calls the classification web service to return the predicted label. The communication with the web services is performed via the standard HTTP request/response mechanism. Fig. 5.17 gives a screenshot of the application. The application can be accessed online via [77].

5.7 Limitations

Even though the study provided promising results, a few limitations should be considered carefully as follows. The main limitation is the relatively small number of participants. Equally important, the lack of a benchmark dataset in the ASD literature makes it difficult to objectively compare our results to other ML approaches.

Another concern could be the duration of videos, which were relatively short. Perhaps longer scenarios would allow for a richer representation and analysis of the gaze behavior. Also, the inclusion of the ADI and ADOS scores could provide further interpretation of the results, especially with respect to the cluster analysis.

5.8 Conclusions

A set of implications can be considered in view of the results experimented in this study. First, the ML experiments confirmed the core idea behind our approach, which hinges on the visual representation of scanpaths. On the one hand, the clustering experiments confirmed the applicability of the visual-based clustering using eye-tracking scanpaths. This could translate into that scanpath visualizations could largely discriminate the ASD samples from others. Furthermore, the cluster analysis could highlight possible connections between the dynamics of eye gaze and autism. It is conceived that the clusters could serve as a kernel for further analysis to bring interesting insights in the context of autism or related applications. Furthermore, the study presented a practical application of deep autoencoders to embed high dimensional data into a lower dimensional latent space. In our case, the embedding space were utilized to learn new representations of the scanpath images, which could encode the most relevant information, and helped reduce sparsity and noise in the original data.

The model performance indicated that the scanpath visualizations could successfully pack the information of gaze movement and its inherent dynamics. This generally demonstrates the validity of applying such visual patterns to detect ASD-diagnosed individuals. To conclude, the coupling of ML methods and eye-tracking holds promising potentials for the development of data-driven techniques to assist the diagnosis of ASD.

References

- [1] L. Wing, J. Gould, Severe impairments of social interaction and associated abnormalities in children: epidemiology and classification, *Journal of Autism and Developmental Disorders* 9 (1) (1979) 11–29.
- [2] A.V. Buescher, Z. Cidav, M. Knapp, D.S. Mandell, Costs of autism spectrum disorders in the United Kingdom and the United States, *JAMA Pediatrics* 168 (8) (2014) 721–728.
- [3] R. Carette, M. Elbattah, G. Dequen, J. Guérin, F. Cilia, J. Bosche, Learning to predict autism spectrum disorder based on the visual patterns of eye-tracking scanpaths, in: *Proc. of the 12th International Conference on Health Informatics*, 2019.
- [4] M. Elbattah, R. Carette, G. Dequen, J. L. Guérin, F. Cilia, Learning clusters in autism spectrum disorder: image-based clustering of eye-tracking scanpaths with deep autoencoder, in: *Proc. of the 41st Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, IEEE, Piscataway, NJ, 2019, pp. 1417–1420.
- [5] E. Schopler, R.J. Reichler, R.F. DeVellis, K. Daly, Toward objective classification of childhood autism: childhood autism rating scale (CARS), *Journal of Autism and Developmental Disorders* 10 (1) (1980) 91–103.
- [6] C. Lord, M. Rutter, A. Le Couteur, Autism diagnostic interview-revised: a revised version of a diagnostic interview for caregivers of individuals with possible pervasive developmental disorders, *Journal of Autism and Developmental Disorders* 24 (5) (1994) 659–685.
- [7] C. Lord, M. Rutter, S. Goode, J. Heemsbergen, H. Jordan, L. Mawhood, et al., Autism diagnostic observation schedule: a standardized observation of communicative and social behavior, *Journal of Autism and Developmental Disorders* 19 (2) (1989) 185–212.
- [8] W. Bosl, A. Tierney, H. Tager-Flusberg, C. Nelson, EEG complexity as a biomarker for autism spectrum disorder risk, *BMC Medicine* 9 (1) (2011) 18.

- [9] H.C. Hazlett, M. Poe, G. Gerig, R.G. Smith, J. Provenza, A. Ross, et al., Magnetic resonance imaging and head circumference study of brain size in autism: birth through age 2 years, *Archives of General Psychiatry* 62 (12) (2005) 1366–1376.
- [10] P. Majaranta, A. Bulling, Eye tracking and eye-based human–computer interaction, in: P. Majaranta, H. Aoki, M. Donegan, D.W. Hansen, J.P. Hansen, A. Hyrskykari, K.J. Rähkä (Eds.), *Gaze Interaction and Applications of Eye Tracking: Advances in Assistive Technologies*, IGI-Gloal, Hershey, 2014, pp. 39–65.
- [11] L. Javal, Essai sur la physiologie de la lecture, *Annales d’Oculistique* 80 (1878) 240–274.
- [12] L. Javal, Essai sur la physiologie de la lecture, *Annales d’Oculistique* 82 (1879) 242–253.
- [13] J.M. Henderson, Human gaze control during real-world scene perception, *Trends in Cognitive Sciences* 7 (11) (2003) 498–504.
- [14] R. Jacob, Eye tracking in advanced interface design, in: W. Barfield W, T.A. Furness (Eds.), *Virtual Environments and Advanced Interface Design*, Oxford University Press, New York, 1995, pp. 258–288.
- [15] E.B. Huey, *The Psychology and Pedagogy of Reading*, The Macmillan Company, New York, 1908.
- [16] G.T. Buswell, *Fundamental Reading Habits: A Study of Their Development*, American Psychological Association, Worcester, MA, 1922.
- [17] G.T. Buswell, *How People Look at Pictures: a Study of the Psychology and Perception in Art*, University of Chicago Press, Chicago, IL, 1935.
- [18] R.N. Khushaba, C. Wise, S. Kodagoda, J. Louviere, B.E. Kahn, C. Townsend, Consumer neuroscience: assessing the brain response to marketing stimuli using electroencephalogram (EEG) and eye tracking, *Expert Systems with Applications* 40 (9) (2013) 3803–3812.
- [19] M.L. Mele, S. Federici, Gaze and eye-tracking solutions for psychological research, *Cognitive Processing* 13 (1) (2012) 261–265.
- [20] S. Khalighy, G. Green, C. Scheepers, C. Whittet, Quantifying the qualities of aesthetics in product design using eye-tracking technology, *International Journal of Industrial Ergonomics* 49 (2015) 31–43.
- [21] A. Vabalas, M. Freeth, Brief report: patterns of eye movements in face to face conversation are associated with autistic traits: evidence from a student sample, *Journal of Autism and Developmental Disorders* 46 (1) (2016) 305–314.
- [22] K. Pierce, D. Conant, R. Hazin, R. Stoner, J. Desmond, Preference for geometric patterns early in life as a risk factor for autism, *Archives of General Psychiatry* 68 (1) (2011) 101–109.
- [23] W. Jones, A. Klin, Attention to eyes is present but in decline in 2–6-month-old infants later diagnosed with autism, *Nature* 504 (7480) (2013) 427–431.
- [24] T.W. Frazier, E.W. Klingemier, M. Beukemann, L. Speer, L. Markowitz, S. Parikh, et al., Development of an objective autism risk index using remote eye tracking, *Journal of the American Academy of Child & Adolescent Psychiatry* 55 (4) (2016) 301–309.
- [25] T.W. Frazier, E.W. Klingemier, S. Parikh, L. Speer, M. S. Strauss, C. Eng, et al., Development and Validation of objective and quantitative eye tracking 2 based measures of autism risk and symptom levels, *Journal of the American Academy of Child & Adolescent Psychiatry* 57 (11) (2018) 858–866.
- [26] A.L. Samuel, Some studies in machine learning using the game of checkers, *IBM Journal of Research and Development* 3 (3) (1959) 210–229.
- [27] P. Covington, J. Adams, E. Sargin, Deep neural networks for YouTube recommendations, In: *Proc. of the 10th ACM Conference on Recommender Systems*, ACM, New York, 2016, pp. 191–198.
- [28] C. Chen, P. Zhao, L. Li, J. Zhou, X. Li, M. Qiu, Locally connected deep learning framework for industrial-scale recommender systems, In: *Proc. of the 26th International Conference on World Wide Web*, 2017, pp. 769–770.
- [29] J.W. Kim, B.H. Lee, M.J. Shaw, H.L. Chang, M. Nelson, Application of decision-tree induction techniques to personalized advertisements on internet storefronts, *International Journal of Electronic Commerce* 5 (3) (2001) 45–62.
- [30] K.W. Cheung, J.T. Kwok, M.H. Law, K.C. Tsui, Mining customer product ratings for personalized marketing, *Decision Support Systems* 35 (2) (2003) 231–243.
- [31] S.F. Weng, J. Reys, J. Kai, J.M. Garibaldi, N. Qureshi, Can machine-learning improve cardiovascular risk prediction using routine clinical data? *PLOS One* 12 (4) (2017) e0174944.
- [32] K. Shameer, M.A. Badgeley, R. Miotto, B.S. Glicksberg, J.W. Morgan, J.T. Dudley, Translational bioinformatics in the era of real-time biomedical, health care and wellness data streams, *Briefings in Bioinformatics* 18 (1) (2017) 105–124.
- [33] R. Agrawal, T. Imieliński, A. Swami, Mining association rules between sets of items in large databases, in: *Proc. of the 1993 ACM International Conference on Management of Data (SIGMOD)*, ACM, New York, 1993, pp. 207–216.
- [34] R. Agrawal, R. Srikant, Fast algorithms for mining association rules, in: *Proc. of the 20th International Conference of Very Large Data Bases (VLDB)*, vol. 1215, 1994, pp. 487–499.

- [35] X. Zhai, A. Oliver, A. Kolesnikov, L. Beyer, S4I: Self-supervised semi-supervised learning, in: Proc. of the IEEE International Conference on Computer Vision (ICCV), IEEE, Piscataway, NJ, 2019, pp. 1476–1485.
- [36] A. Kolesnikov, X. Zhai, L. Beyer, Revisiting self-supervised visual representation learning, in: Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, Piscataway, NJ, 2019, pp. 1920–1929.
- [37] G. Pusioli, A. Esteve, S. S. Hall, M. Frank, A. Milstein, L. Fei-Fei, Vision-based classification of developmental disorders using eye-movements, in: Proc. of the International Conference on Medical Image Computing and Computer-Assisted Intervention, Springer, Cham, 2016, pp. 317–325.
- [38] R. Carette, F. Cilia, G. Dequen, J. Bosche, J.L. Guerin, L. Vandromme, Automatic autism spectrum disorder detection thanks to eye-tracking and neural network-based approach, Proceedings of the International Conference on IoT Technologies for HealthCare, Springer, 2017, pp. 75–81.
- [39] V. Yaneva, L. A. Ha, S. Eraslan, Y. Yesilada, R. Mitkov, Detecting autism based on eye-tracking data from web searching tasks, in: Proceedings of the Internet of Accessible Things, ACM, New York, 2018, pp. 16.
- [40] A. Anzulewicz, K. Sobota, J.T. Delafeld-Butt, Toward the autism motor signature: gesture patterns during smart tablet gameplay identify children with autism, Scientific Reports 6 (1) (2016) 1–13.
- [41] P. Geurts, D. Ernst, L. Wehenkel, Extremely randomized trees, Machine Learning 63 (1) (2006) 3–42.
- [42] L. Breiman, Random forests, Machine Learning 45 (1) (2001) 5–32.
- [43] R. Johnson, T. Zhang, Learning nonlinear functions using regularized greedy forest, IEEE Transactions on Pattern Analysis and Machine Intelligence 36 (5) (2013) 942–954.
- [44] M. Jiang, S. M. Francis, D. Srishyla, C. Conelea, Q. Zhao, S. Jacob, Classifying individuals with ASD through facial emotion recognition and eye-tracking, in: Proc. of the 41st Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), IEEE, Piscataway, NJ, 2019, pp. 6063–6068.
- [45] S.W. Porges, J.F. Cohn, E. Bal, D. Lamb, The dynamic affect recognition evaluation, Brain-Body Center, University of Illinois at Chicago, Chicago, IL, 2007.
- [46] Y. LeCun, Y. Bengio, G. Hinton, Deep learning, Nature 521 (7553) (2015) 436–444.
- [47] Y. LeCun, B. E. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. E. Hubbard, et al., Handwritten digit recognition with a back-propagation network, in: Proc. of Advances in Neural Information Processing Systems (NIPS), 1990, pp. 396–404.
- [48] Y. LeCun, L. Bottou, Y. Bengio, P. Haffner, Gradient-based learning applied to document recognition, Proceedings of the IEEE 86 (11) (1998) 2278–2324.
- [49] B.A. Pearlmutter, Learning state space trajectories in recurrent neural networks, Neural Computation 1 (2) (1989) 263–269.
- [50] A. Krizhevsky, I. Sutskever, G. E. Hinton, ImageNet classification with deep convolutional neural networks, in: Proc. of Advances in Neural Information Processing Systems (NIPS), 2012, pp. 1097–1105.
- [51] J. Gehring, M. Auli, D. Grangier, D. Yarats, Y. N. Dauphin, Convolutional sequence to sequence learning, in: Proc. of the 34th International Conference on Machine Learning (ICML), vol. 70, 2017, pp. 1243–1252.
- [52] S. Chen, Q. Zhao, Attention-based autism spectrum disorder screening with privileged modality, in: Proc. of the IEEE International Conference on Computer Vision (ICCV), IEEE, Piscataway, NJ, 2019, pp. 1181–1190.
- [53] M. Jiang, Q. Zhao, Learning visual attention to identify people with autism spectrum disorder, in: Proc. of the IEEE International Conference on Computer Vision (ICCV), IEEE, Piscataway, NJ, 2017, pp. 3267–3276.
- [54] S. Ozonoff, B.L. Goodlin-Jones, M. Solomon, Evidence-based assessment of autism spectrum disorders in children and adolescents, Journal of Clinical Child and Adolescent Psychology 34 (3) (2005) 523–540.
- [55] J. H. Goldberg, J. I. Helfman, Visual scanpath representation, in: Proc. of the 2010 Symposium on Eye-Tracking Research & Applications, ACM, New York, 2010, pp. 203–210.
- [56] D. Noton, L. Stark L., Scanpaths in eye movements during pattern perception, Science 171 (3968) (1971) 308–311.
- [57] D. Noton, L. Stark L., Scanpaths in saccadic eye movements while viewing and recognizing patterns, Vision Research 11 (9) (1971) 929–942.
- [58] J.D. Hunter, Matplotlib: A 2D graphics environment, Computing in Science & Engineering 9 (3) (2007) 90–95.
- [59] Figshare, Visualization of eye-tracking scanpaths in autism spectrum disorder: image dataset. , https://figshare.com/s/5d4f93395cc49d01e2bd_, 2018 (accessed 07.06.22).
- [60] K.L. Wu, M.S. Yang, Alternative c-means clustering algorithms, Pattern Recognition 35 (10) (2002) 2267–2278.
- [61] P.S. Bradley, U.M. Fayyad, Refining initial points for K-Means clustering, in: Proceedings of the 15th International Conference on Machine Learning (ICML), 1998, pp. 91–99.

- [62] L.V.D. Maaten, G. Hinton, Visualizing data using t-SNE, *Journal of Machine Learning Research* 9 (2008) 2579–2605.
- [63] F. Chollet, *Deep Learning with Python*, Manning Publications Co, Shelter Island, NY, 2017.
- [64] D.E. Rumelhart, G.E. Hinton, R.J. Williams, Learning internal representations by error propagation, *Parallel Distributed Processing*, vol. 1, Foundations, MIT Press, Cambridge, MA, 1986.
- [65] Y. Wang, H. Yao, S. Zhao, Auto-encoder based dimensionality reduction, *Neurocomputing* 184 (2016) 232–242.
- [66] M. Sakurada, T. Yairi, Anomaly detection using autoencoders with nonlinear dimensionality reduction, in: *Proc. of the MLSDA 2014 2nd Workshop on Machine Learning for Sensory Data Analysis*, 2014, pp. 4–11.
- [67] D.P. Kingma, J. Ba, Adam: a method for stochastic optimization, in: *Proc. of the 3rd International Conference on Learning Representations (ICLR)*, 2015.
- [68] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, R. Salakhutdinov, Dropout: a simple way to prevent neural networks from overfitting, *Journal of Machine Learning Research* 15 (1) (2014) 1929–1958.
- [69] F. Chollet, Keras. , <https://github.com/fchollet/keras> . , 2015 (accessed 07.06.22).
- [70] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, et al., Tensorflow: a system for large-scale machine learning, in: *Proceedings of the 12th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 16)*, 2016, pp. 265–283.
- [71] A.K. Jain, Data clustering: 50 years beyond K-means, *Pattern Recognition Letters* 31 (8) (2010) 651–666.
- [72] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, et al., Scikit-learn: machine learning in python, *Journal of Machine Learning Research* 12 (2011) 2825–2830.
- [73] P.J. Rousseeuw, Silhouettes: a graphical aid to the interpretation and validation of cluster analysis, *Journal of Computational and Applied Mathematics* 20 (1987) 53–65.
- [74] G. Bradski, The OpenCV library, *Dr Dobbs's Journal of Software Tools* 25 (2000) 120–125.
- [75] Y. Xu, R. Jia, L. Mou, G. Li, Y. Chen, Y. Lu, et al., Improved relation classification by deep recurrent neural networks with data augmentation, 2016. Available from: , <https://doi.org/10.48550/arXiv.1601.03651> . .
- [76] L. Perez, J. Wang, The effectiveness of data augmentation in image classification using deep learning, 2017. Available from: , <https://arxiv.org/pdf/1712.04621.pdf> . .
- [77] URL: <https://goo.gl/i4N7Zj>. (accessed 07.06.22).