Diagnosing Autism Using Eye-Tracking Data and Ensemble Learning

Brijesh Prajapati [216066072]

University of Johannesburg, Johannesburg, South Africa

Abstract. ASD is a neurodevelopmental disorder in which an individual shows a lack of social and communication skills as well as a reduced attention span. The impact of ASD can be reduced to a great extent on an individual with early diagnosis. One of the problems with clinical diagnosis is that they are expensive and take a lot of time for an accurate diagnosis. In this article, we present an application that can use eye-tracking data collected from free image viewing to classify the scan paths as belonging to an ASD child or a typically developing child. This approach can be used by parents to assess if their child has ASD.

Keywords: Autism Spectrum Disorder, ensemble learning, SVM, Random Forest, MLP, AdaBoost

1 Introduction

Autism Spectrum Disorder (ASD) is a neurodevelopmental disorder that is evident in individuals with deficits in social and communication skills, repetitive actions, body control, and a host of other distinct conditions. Individuals diagnosed with ASD have found to have diminished focus and social perception of their environment since early childhood. Some of these disabilities and impairments vary among individuals depending on the severity of the autism.

Clinical practitioners diagnosing individuals with ASD rely primarily on observing and interviewing the individuals. Some of the aspects that clinical practitioners perceive include attentiveness of individuals to their name-calling, the scanning patterns of individuals when looking at a new person for the first time, the ability to recognize emotions, and identifying faces from an image. These observations to diagnose ASD is time-consuming, complicated, expensive, and primarily based on the practitioner's interpretations. Moreover, these observations can only be conducted by specialized medical practitioners in specialized medical centers. Sometimes it even happens that the diagnosis can differ from practitioners to practitioners because there are no significant observable differences between the ASD and normal individuals leading to inaccuracies in diagnosis.

Early and accurate diagnosis of ASD is highly beneficial for individuals as well as their families because there is a higher chance that initial treatments can significantly reduce the impact of ASD on an individual's life and expensive medical bills [1]. Developing an application using machine learning techniques that will allow parents to conduct an initial ASD evaluation test on their children if they are skeptical of their child's communication and social skills to gauge if they have ASD. An application-based approach to ASD diagnosis will allow parents in locations where there are no ASD medical clinics available for diagnosis to diagnose their children before consulting an expensive clinical practitioner. The results from the applications can also be used by medical practitioners to gauge the intensity of autism in children [4].

Eye movements provide a great source of information on the attention span and scanning patterns of an individual when viewing different images [2]. And since individuals with ASD have different attention span compared to normal individuals, we can use this information for ASD diagnosis. For the solution to diagnose ASD, we will be using eye-tracking data of children from freely viewing a collection of images, and the eye-tracking data is collected using an eye-tracker. The eye-tracking data is used to extract features that provide insightful information that can be used to perform classification on the new eye-tracking data. The proposed solution uses a heterogeneous ensemble machine learning approach for training models and then using the trained models to classify new eye-tracking data.

In this research paper, some of the current solutions that use eye-tracking data for autism classification are evaluated. The research paper explains the dataset obtained and then introduces the proposed solution and the importance of the various features derived from the dataset. Furthermore, the research paper discusses the choice of classifiers used for the ensemble learning model. The research paper then presents the results and accuracies for various models for different features sets and as well as the overall accuracy of the ensemble learning. And finally, the research paper concludes the findings of the proposed solution and future improvements to increase accuracy.

2 Literature Review

Accurate eye-tracking devices, along with advances in machine learning algorithms, have enhanced the diagnostic accuracy of individuals with autism as well as other disorders that can be diagnosed by monitoring eye-tracking data. One of the approaches presented in [3] uses a visual representation of eye-tracking scan paths to differentiate between children with autism and typically developing. They use eye-tracking data collected to create a color encoded scan path visualization map which, is then gray scaled and scaled-down to reduce the number of features. They then extract pixel values for each scan path to be used as their primary feature. For developing their classification model, they venture different algorithms such as Naive Bayes, Support Vector Machine, Logistic Regression, and Neural Networks to test which algorithms provide the highest accuracy from the derived features. In their experimental results, the neural network model with one layer of fifty neurons achieved the highest classification accuracy based on the derived features. One of the disadvantages of the approach outlined in [3] is that the same video used to collect eye-tracking data for training should be displayed if a new child needs to be diagnosed.

A different method proposed in [5] to diagnose children with autism is based on the fact that children with ASD have a diminished perception of human faces, which leads to atypical face processing. The solution in [5] uses machine learning to analyze face scanning patterns of children looking at images of human faces to diagnose them for autism. The features derived from scanning data collected of children looking at various human faces include the bag-of-words on eye-gaze coordinates, which uses the centers of concentrated visual attention as dictionary words to generate a frequency distribution of gaze on different parts of the face image. Another feature derived is bag-of-words on eye-motion in which a motion vector of eye-gaze is calculated to obtain the bag-of-words histogram. Using the derived features support vector machine with radial basis function (RBF) kernel is trained and tested and achieved an overall accuracy of 0.92. One of the drawbacks of this approach is to diagnose a child with autism, a very particular type of images, images of human faces need to be shown, which can sometimes be challenging to obtain.

Similar to [4] and [5], another method proposed by [6] uses stimuli that include a brief video of a human face to obtain eye-tracking data from children between 4-6 years of age with ASD and typically developing. The goal of the research was to identify areas of interest on the face that could be used to differentiate eye-tracking data for children with ASD and without ASD. In their observations, by performing discriminant analysis, they found that the fixation times at the eyes, nose, mouth, persons' face, and body were notably lower in children with ASD. Based on their findings, they use a support vector machine for classifying children with and without ASD using the predefined AOIs and the ratios of the fixation time to different areas of the video.

3 Proposed System Model

The system proposed in this research takes into consideration some of the drawbacks of the previously discussed approaches and presents an effort to improve the diagnosis approach of children with autism based on eye-tracking data obtained from viewing any static image. The proposed method uses scan path features and saliency features combined with ensemble machine learning to diagnose children with autism.

Figure 1 illustrates the pipeline used by the system for training the models from the acquired dataset. In the first process, the obtained training dataset is loaded into memory for feature extraction and training. The second process, named Separate Scan path, splits the scan path file into individual scan paths, which can then be used for computing the scan path and saliency features. To extract the saliency features saliency map is required, which is created by the compute saliency function available in the OpenCV library. In the fourth process, some data-processing such as normalizing the feature values is applied for the support vector machine and the multilayer perceptron to reduce data redundancy and improve the accuracy of the model. In the fifth process, different machine learning algorithms such as the linear SVM, Random Forest, Multilayer perceptron, and Adaboost are combined to create an ensemble learning model. And in the final part, these algorithms are trained on the training data, and for testing new data, a voting-based approach is used for the final classification.

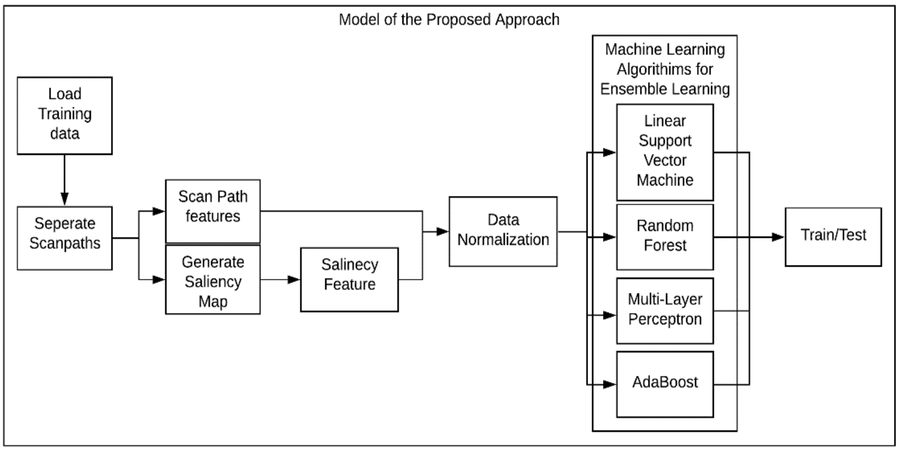

Figure 1. The proposed approach of the system.

4 Implementation of the Proposed System

4.1 Dataset Acquisition

The data set used to extract features and train the ensemble model was obtained from [7]. The collected dataset consists of 300 images and the corresponding scan-paths of 14 different children with ASD and typically developing having an average age of 8 years [8]. The 300 images selected by the researchers were of various types, such as images containing natural scenes, different types of animals, multiple objects, and single or multiple people.

For the experiment, the researchers used an eye-tracker to display the images and record the eye movements. The displayed images were selected at random and each image was displayed for 3 seconds with a 1-second interval between each image [8]. The recorded eye-movements are given in the form of x and y coordinates with the duration of each fixpoint. The obtained dataset also gives a saliency map of each image. The saliency map is generated by combining all the fixpoints for that image and applying a gaussian blur.

4.2 UML Class Diagram of the Proposed System

To demonstrate that the proposed system can work to diagnose children with autism based on eye-tracking data, this section of the research provides insight into the functioning of the most fundamental components of the proposed application. This section introduces a system class diagram in figure 2, which is then utilized to explain how the system operates to perform classifications. In the following sections, the research paper describes the components of the system.

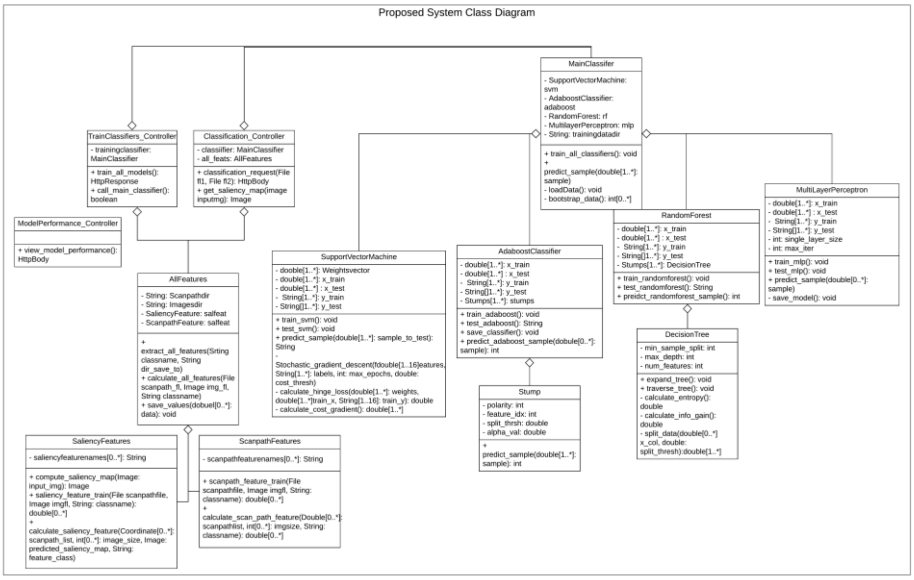

Figure 2. Illustrating the class diagram of the proposed system which that diagnose children with autism or normal.

4.3 Features Extracted from the Dataset

The features examined in this research paper are considered to classify scan paths as belonging to an individual with or without ASD. Two primary feature sets, the scan path, and saliency features are derived, from the scan path fixation points and their corresponding stimulus image provided in the dataset. From the class diagram in figure 2, there are two different classes created for deriving each of the features separately, which are then combined in the class named AllFeatures forming, an aggregation relationship between the two system classes.

The scan path features are derived using the x and y coordinates, the duration at each fixation point, and the dimensions of the image displayed to capture the scan paths. Children with autism have a variety of attentional variations, such as decreased social and joint attention behaviors compared to typically developing children [9]. Scan path features used to learn to differentiate between a scan path belonging to a normal child or child with ASD include fixpoint count, total duration, mean duration, total scan path length, average scan path length, the average distance from the center of the image, and the average distance from the average coordinate. These features are calculated in the calculation scan path feature method in the scan path class from the received scan path and the image dimension as shown in figure 2.

The saliency features are extracted using the saliency map created of the image shown to capture the scan paths. Saliency features are relevant because children with ASD have an unusual saliency to different image stimuli. The saliency features are derived from the saliency values extracted from the generated saliency map using the provided scan path coordinates. Characteristics derived from the list of saliency values obtained from the saliency map include the value of the first saliency fixation, average weighted duration, maximum saliency value, and the normalized saliency scan path. These features are derived in the calculate saliency feature method in the saliency features class that receives the saliency map of the image along with the scan path list as a parameter and returns a list of the calculated feature values.

The All Features class creates the Scanpath Feature and Saliency Feature objects to derive both types of features and combine them into a single feature value list. This class is also responsible for saving the combined feature value list along with the label (ASD or TD) into a comma-separated values file, which is more manageable for the training models to read and train.

4.4 Implementation of the Ensemble Learning

Ensemble learning is a machine learning method that incorporates the decisions of several classifiers to make a final decision. The proposed system in this research paper makes use of supervised heterogeneous ensemble learning to perform its final classification of scan paths. Ensemble learning used in the proposed solution is referred to as supervised heterogeneous because the training data collected has labels, and it is heterogeneous because different machine learning algorithms are used instead of a single type. The system class diagram in figure 2 illustrates that the Main Classifier class incorporates different classifiers to behave as one final classifier.

The Main Classifier class in the class diagram illustrated in figure 2 is responsible for training each of the algorithms using bootstrapping the samples. Bootstrapping is a data resampling technique where the training data is shuffled randomly with replacement, meaning the same record can appear more than once in the training data. Bootstrapping is used to gauge the skill of each of the machine learning algorithms when predicting new data [10]. For testing and classification, the Main Classifier uses a voting-based approach for classification. The most common label classified by the different classifiers is the final prediction. In the case of a tie, the Main Classifier default the classification to ASD because if there is a 50% chance of the scan path belonging to a child with ASD, it is safe to say that the child might have ASD.

The machine learning algorithms that make up the final ensemble classifier make use of bagging and boosting methods using random forest and Adaboost, respectively. Bagging is a type of ensemble learning method in which multiple classifiers of the same kind with high variances, such as decision trees, are trained using the same data samples by bootstrapping [11]. Boosting is also another type of ensemble learning method in which multiple week classifiers are combined to form one strong classifier. In boosting, misclassified data by the previous weak classifier are advanced to the next weak classifier to improve on those misclassifications. The final ensemble classifier also makes use of the support vector machine and multi-layer perceptron (MLP). To improve the performance of the SVM and MLP classifiers, data normalization is used to normalize the feature values because the feature values of children with ASD vary drastically from normal children. So, data normalization improves performance by scaling down the feature values in the range between 0 and 1.

The main classifier class creates objects of different classifiers to train, test, and perform predictions. The Main Classifier is also responsible for converting string labels into integer labels (1: ASD, -1: Normal) and then loading the feature values into memory so that the sample list can be passed to the classifiers for training after bootstrapping the training data samples. It also manages the final predictions on new data by using a voting-based approach to give the final prediction.

4.5 Final Demo Application

The final demo application is a web application, developed using the Asp.net framework, which can be accessed by the users to perform classification on new data as well as train models by specifying the parameters of each algorithm. The web app also allows the user to view the results of the trained models based on their specified parameters used for training.

The web app requests the controller of the web application program interface (API) developed using the flask framework. As illustrated in figure 2 of the class diagram, the web application makes requests to the controller classes based on the data it wants to process or read. For instance, if the user of the web app requests to train the models, then the web app will call the training method in the Train Classifier Controller class. By using this distributed approach, users do not require powerful hardware for training machine learning models. Users can easily upload eye-tracking data and receive the classification report.

5 Results

This section of the research presents the results of different classifiers in the form of various metrics that help to evaluate the ability of each classifier as well as the final classifier to diagnose children with ASD based on eye-tracking data. The result of the final classifier is presented in figure 3.

Predicted/Actual	Predicted Positive	Predicted Negative	
Actual Positive	TP: 157	FN: 93	Sensitivity: 0.628
Actual Negative	FP: 68	TN: 207	Specificity: 0.753
	Precision: 0.698	Negative Prediction Value: 0.690	Accuracy: 0.693

Figure 3. Displaying the report of the final classifier.

The performances of the different classifiers are compared in figure 4 is based on tests conducted on 525 samples after training, which is 20% of the total dataset used for testing. The system uses a confusion matrix along with different metrics to evaluate and compare the performances of each classifier. The confusion matrix consists of true positive (TP), false negative (FN), false positive (FP), and true negatives (TN) values. The TP number shows the number of ASD samples that were correctly classified, TN number displays the number of healthy samples that were correctly classified. Whereas FN presents the number of samples that were ASD and were misclassified by the model, and FP displays the number of samples that were normal but misclassified as belonging to ASD subject.

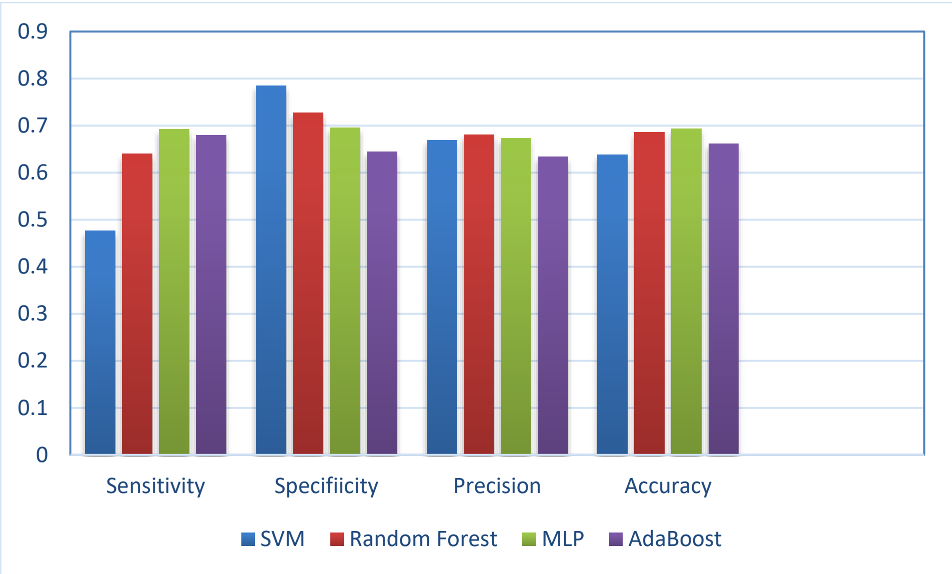

Figure 4. A bar graph displaying side by side comparison of sensitivity, specificity, precision, and accuracy of different classifiers.

The parameter values set to train each of the classifiers is as follows: for SVM, maximum epochs were set to 16000 and cost threshold was set to 0.0001. For Adaboost, number of stumps was set to 16. For random forest, number of trees was set to 10 and maximum tree depth was set to 8. For MLP, single layer size was set 16 perceptions and maximum number of iterations over the training data was set to 400.

The sensitivity metric gives the TP rate, and specificity gives TN valuation. Precision shows the ASD classifications ratio, and negative prediction shows the Normal classifications ratio. And finally, the accuracy shows the rate of the total number of classifications that were correctly classified. These metrics indicate how well each of the models is going to perform on new data. By evaluating these metrics, it can be determined how effective the models are on new data sent by users.

6 Critique/Analysis of the Proposed System

Evaluating the results shown in Figure 3 and Figure 4, the final classifier has achieved an accuracy of 69.3 percent, suggesting that the final classifier is 69.3 percent likely to correctly diagnose new data. However, it is also shown that the final model achieved a higher specificity rate of 75.3 percent compared to a sensitivity rate of 62.8 percent. This indicates that the final classifier is more likely to correctly identify normal children than children with ASD.

Moreover, evaluating the comparisons of different classifiers in the bar graph in figure 4 it can be observed that the MLP and random forest achieved a high accuracy compared to SVM and Adaboost. Another observation that can be made is that SVM is most likely to correctly identify normal children compared to other classifiers, and it also least likely to correctly classify children with ASD. This indicates that linear SVM might not be the best fitting classifier to use for ASD diagnosis based on eye-tracking data. It is also observed that other classifiers can be improved by performing a parameter study.

The decision to use the random forest, MLP and Adaboost classifiers was the right choice as they had higher overall accuracy compared to SVM. And by using these three classifiers in the final model, the overall accuracy of the final classifier would be greatly improved. It is also possible to improve the accuracy of each model by playing with the parameter values of each of these classifiers, thereby improving the overall accuracy of the final model.

Moreover, from the results, it can be observed that children with mild symptoms of autism have very similar scan-path data compared to normal children resulting in lower accuracy of the models. To further improve the system as well as accuracy, a feature study can be conducted in which reducing the number of features to only those features with the high variance between scan-paths belonging to ASD or normal child.

To further improve the system, a weight-based method can be used for the final classification. This approach works by initializing a weight of 1 for each classifier, and if a particular classifier misclassifies, the weight associated with that classifier decrease by a certain amount. And once the weight reaches below a certain threshold the classification result of that classifier will not be considered in the final classification. By using this approach weak classifiers such as the linear SVM will be removed from the list of classifiers and therefore improving the accuracy of the final model.

By implementing some of the changes stated in the previous paragraphs, the system can significantly improve the accuracy of the models. The use of ensemble learning and choices of classifiers, except for the linear SVM, has proven to be a reliable choice. Overall, the use of ensemble learning has proven to be a useful approach in this scenario of diagnosing children with autism using eye-tracking data.

7 Conclusion

This research paper suggested a method that could use eye-tracking data to diagnose children with autism using ensemble machine learning to enhance the accuracy of the final model. This research paper explored different scan-paths and derived saliency features, which were then used to train different classifiers by bootstrapping samples.

The proposed system does not achieve high accuracy, but it shows how complex neurodevelopmental disorders such as autism can be diagnosed using eye-tracking data and an ensemble learning approach. With some of the improvements stated, the accuracy of the system can be significantly improved.

References

1. Fletcher-Watson, S., Leekam, S., Benson, V., Frank, M., Findlay, J.: Eye-movements reveal attention to social information in autism spectrum disorder. Neuropsychologia. 47, 248-257 (2009).
2. Startsev, M., Dorr, M.: Classifying Autism Spectrum Disorder Based on Scanpaths and Saliency. 2019 IEEE International Conference on Multimedia & Expo Workshops (ICMEW). (2019).
3. Carette, R., Elbattah, M., Cilia, F., Dequen, G., Guérin, J., Bosche, J.: Learning to Predict Autism Spectrum Disorder based on the Visual Patterns of Eye-tracking Scanpaths. Proceedings of the 12th International Joint Conference on Biomedical Engineering Systems and Technologies. (2019).
4. Duan, H., Zhai, G., Min, X., Fang, Y., Che, Z., Yang, X., Zhi, C., Yang, H., Liu, N.: Learning to Predict where the Children with Asd Look. 2018 25th IEEE International Conference on Image Processing (ICIP). (2018).
5. Liu, W., Yu, X., Raj, B., Yi, L., Zou, X., Li, M.: Efficient autism spectrum disorder prediction with eye movement: A machine learning framework. 2015 International Conference on Affective Computing and Intelligent Interaction (ACII). (2015).
6. Wan, G., Kong, X., Sun, B., Yu, S., Tu, Y., Park, J., Lang, C., Koh, M., Wei, Z., Feng, Z., Lin, Y., Kong, J.: Applying Eye Tracking to Identify Autism Spectrum Disorder in Children. Journal of Autism and Developmental Disorders. 49, 209-215 (2018).
7. Duan, H., Zhai, G., Min, X., Che, Z., Fang, Y., Yang, X., Gutiérrez, J., Le Callet, P.: A Dataset of Eye Movements for the Children with Autism Spectrum Disorder, https://zenodo.org/record/2647418#.X3eCC2gzZPY.
8. Duan, H., Zhai, G., Min, X., Che, Z., Fang, Y., Yang, X., Gutiérrez, J., Callet, P.: A dataset of eye movements for the children with autism spectrum disorder. Proceedings of the 10th ACM Multimedia Systems Conference. (2019).
9. Wang, S., Jiang, M., Duchesne, X., Laugeson, E., Kennedy, D., Adolphs, R., Zhao, Q.: Atypical Visual Saliency in Autism Spectrum Disorder Quantified through Model-Based Eye Tracking. Neuron. 88, 604-616 (2015).
10. Brownlee, J.: A Gentle Introduction to the Bootstrap Method, https://machinelearningmastery.com/a-gentle-introduction-to-the-bootstrap-method/.
11. Brownlee, J.: Bagging and Random Forest Ensemble Algorithms for Machine Learning, https://machinelearningmastery.com/bagging-and-random-forest-ensemble-algorithms-for-machine-learning/.