

Identification Of Autism In Children Using Static Facial Features And Deep Neural Networks

Ms Afsha Sultana¹, I Meghana², K Anusha³, G Srujathi⁴

¹Assistant Professor; Department Of Computer Science And Engineering(AI&ML) Bhoj Reddy Engineering College For Women Hyderabad India.

^{2,3,4}B.Tech Students; Department Of Computer Science And Engineering(AI&ML) Bhoj Reddy Engineering College For Women Hyderabad India.

Mail Id; imeghana.2026@gmail.com², kuruvaanusha2020@gmail.com³, srujathigudimella3@gmail.com⁴

Abstract

This research presents an intelligent and non-invasive framework for the early identification of Autism Spectrum Disorder (ASD) in children through the analysis of static facial features using deep learning techniques. ASD is a neurodevelopmental disorder that affects communication, social interaction, and behavioural responses, and its symptoms vary significantly among individuals. Early diagnosis is essential because timely therapeutic intervention can greatly improve cognitive development, language skills, and social adaptation. However, conventional diagnostic procedures mainly depend on behavioural observation, interviews, and clinical assessments, which are often time-consuming, subjective, and inconsistent across practitioners. To address these limitations, the proposed study explores the capability of artificial intelligence to support early autism screening through automated facial image analysis.

Introduction

Autism Spectrum Disorder is one of the most common developmental disorders affecting children worldwide. The condition is generally identified through behavioural signs such as delayed speech, repetitive actions, limited eye contact, and difficulty in social communication. Since these symptoms may appear gradually and vary from child to child, early detection remains a challenging task. In many cases, diagnosis occurs later than recommended, reducing the effectiveness of early intervention programs. Therefore, there is a strong need for efficient and objective screening systems that can support healthcare professionals.

Recent developments in artificial intelligence and deep learning have opened new possibilities in medical image analysis. Facial characteristics often contain subtle biological markers associated with developmental conditions. By analysing facial images using advanced neural networks, it becomes possible to identify patterns that are not easily visible to the human eye. This research utilizes Convolutional Neural Networks (CNNs) to automatically learn these features and assist in the early identification of autism in children.

Existing System

Traditional autism diagnosis is primarily based on behavioural assessments, parent questionnaires, developmental checklists, and specialist observations. These methods are widely used in hospitals and psychological centres, but they often require multiple sessions over a long period. The final diagnosis depends greatly on clinical expertise and the child's cooperation during evaluation. As a result, the screening process may become slow, subjective, and inconsistent across different practitioners.

Problems in Existing System

The current diagnostic process faces several practical limitations. It is time-consuming and may delay the start of therapy. Early-stage autism symptoms are sometimes difficult to recognize, leading to missed or late diagnosis. In rural or underdeveloped areas, access to trained specialists is limited. Furthermore, no widely adopted automated system currently exists for analysing facial biomarkers related to autism in routine screening practice.

Proposed System

The proposed system introduces an automated screening framework that uses static facial images of children to detect autism-related traits. A Convolutional Neural Network is trained using labelled facial image datasets containing autistic and non-autistic children. During training, the model learns visual features such as facial symmetry, eye spacing, forehead structure, and other subtle patterns. Once trained, the model predicts whether a new facial image belongs to the autistic or non-autistic class.

To make the solution practical and accessible, the trained model is integrated into a Flask-based web application. Users can upload an image through the interface, and the system provides an instant prediction. This enables real-time preliminary screening in clinics, schools, or home environments.

Advantages of Proposed System

The proposed model offers several benefits over conventional methods. It provides fast and objective predictions without requiring lengthy behavioural observation. The system reduces dependency on specialist availability and can support doctors during preliminary assessments. It is user-friendly, cost-effective, and suitable for large-scale screening programs. Most importantly, it can help identify

children at risk during early developmental stages, enabling timely intervention and better treatment outcomes.

Literature Survey

Previous studies have highlighted the increasing prevalence of Autism Spectrum Disorder and the urgent need for reliable early screening tools. Several healthcare organizations have reported that behavioural screening methods alone may delay accurate diagnosis. Research on questionnaires such as M-CHAT has shown usefulness in early detection, but these approaches depend on caregiver responses and clinician judgement.

Machine learning researchers have proposed various classification models using behavioural and clinical datasets, achieving moderate success. More recently, deep learning techniques have gained attention for image-based diagnosis because of their ability to automatically extract relevant features. Studies involving facial image analysis have shown promising results in identifying autism-related characteristics, demonstrating the effectiveness of CNN-based models. These findings strongly support the methodology used in the present research.

Life Cycle Model

The Agile software development model is adopted for the implementation of this project because of its iterative and flexible nature. Agile supports incremental development, continuous improvement, and quick adaptation to changing requirements. This is especially useful in AI-based systems where model tuning and interface refinement may be required repeatedly. During the requirements phase, the core features such as login, image upload, and prediction modules are identified. In the design phase, the system architecture, database schema, and CNN integration flow are planned. The development phase focuses on implementing modules such as authentication, image preprocessing, and prediction services. Continuous testing is carried out to verify both software functionality and model performance. The deployment phase enables release of usable versions of the application for real-time testing. In the review phase, feedback is collected to improve user experience, model accuracy, and system efficiency in subsequent iterations. Thus, the Agile model provides a suitable framework for developing the proposed intelligent autism screening system.

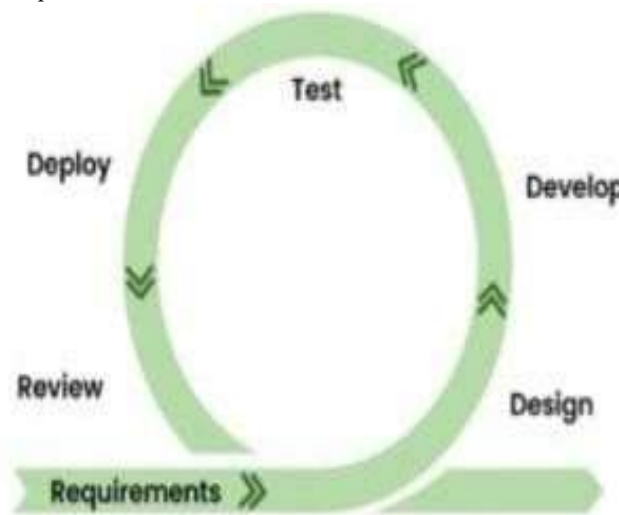


Fig: 1 Agile Model

Design

System design defines the structural and operational blueprint of the proposed application. It explains how various components interact to perform authentication, image processing, prediction, data storage, and result visualization. Proper design ensures maintainability, efficiency, and scalability of the software solution.

Architectures

Project architecture represents the arrangement of system modules and the sequence in which requests are processed. It provides a formal representation of the system structure, helping developers understand component interaction and workflow. The

architecture of the proposed system consists of software architecture and technical architecture.

Software Architecture

The software architecture follows a layered model consisting of presentation layer, application layer, model layer, and database layer. The presentation layer includes the web interface developed using HTML, CSS, and JavaScript. The application layer is managed by Flask, which handles routing, user requests, authentication, and communication with the CNN model.

The model layer contains the trained Convolutional Neural Network responsible for feature extraction and image classification. The database layer uses MySQL to store user details, prediction history, and

report records. Communication among these layers ensures seamless processing from image upload to final result generation.

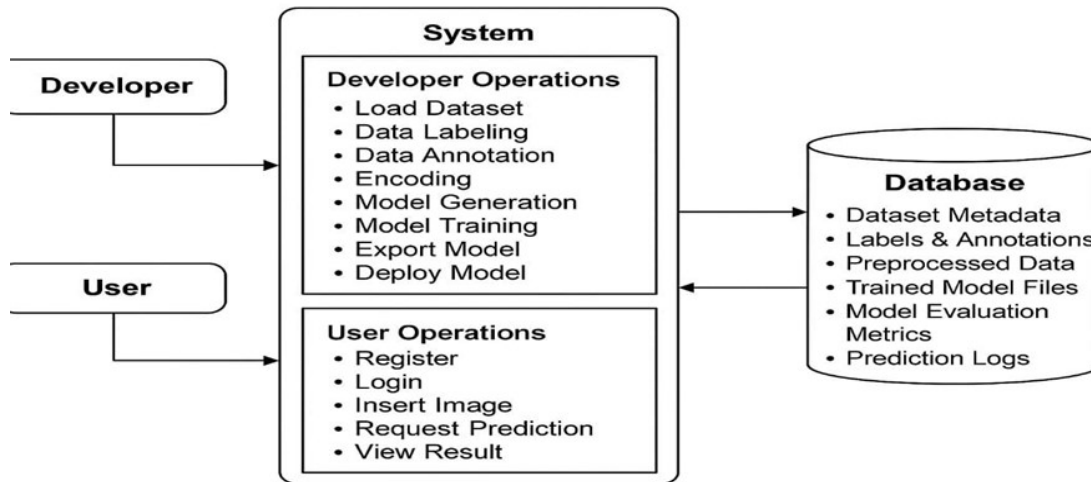


Fig:2 Software Architecture

Technical Architecture

The technical architecture describes the technologies and tools used for deployment and execution. The user accesses the application through a web browser connected to the Flask server. Uploaded images are passed to the preprocessing module, where they are resized and normalized before entering the CNN model. The prediction output is then sent back to the Flask application,

which displays the result along with confidence score.

All user information and screening records are maintained in the MySQL database. Python libraries such as TensorFlow, Keras, NumPy, and OpenCV are used for model execution and image handling. This architecture ensures smooth integration of web technologies, machine learning components, and database management for real-time autism screening.

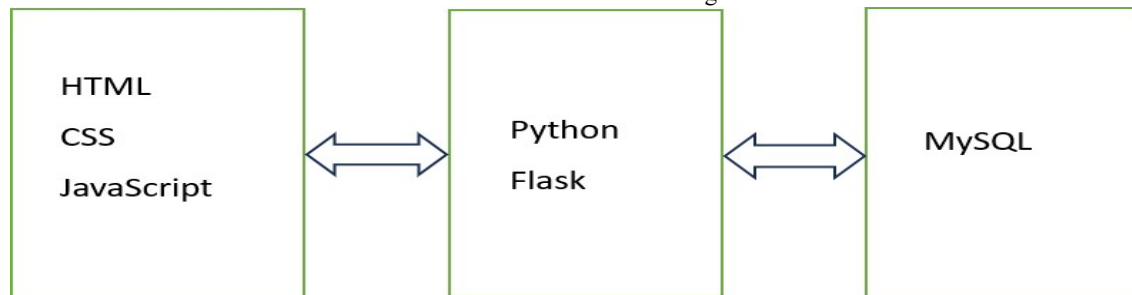


Fig:3 Technical Architecture

Implementation

The implementation phase focuses on transforming the proposed design into a functional application capable of detecting Autism Spectrum Disorder (ASD) using static facial images and deep learning. The system was developed using Python, TensorFlow, Flask, and SQLite. It combines machine learning model training with a web-based deployment platform for real-time prediction and secure user access.

Data Collection and Loading

Facial image data of autistic and non-autistic children were collected from publicly available datasets. The dataset was imported into the Python environment using Google Colab for training,

experimentation, and model development. Images were arranged into two primary folders named **Autistic** and **Non_Autistic**. This directory-based structure enabled the framework to automatically assign class labels during training using image generators.

Data Labeling and Annotation

Each image in the dataset was assigned a class label according to its category. Images belonging to autistic children were labeled as **0**, while images of non-autistic children were labeled as **1**. Since the problem is an image classification task, complex annotation such as bounding boxes or segmentation masks was not required. The complete facial image was used as the input sample for classification.

Data Preprocessing

Before training the CNN model, all images were preprocessed to maintain consistency and improve learning performance. The preprocessing stage included reading images through OpenCV, resizing them to a fixed resolution of **256 × 256 pixels**, normalizing pixel intensities to a range between 0 and 1, and converting them into tensor format suitable for TensorFlow processing. These steps reduced computational complexity and ensured standardized input dimensions.

Exploratory Data Analysis

Exploratory Data Analysis (EDA) was conducted to understand the characteristics of the dataset prior to model training. This included examining the number of images in each class, visualizing sample images, checking image resolutions, and identifying class imbalance issues. Such analysis helped in selecting suitable preprocessing and augmentation strategies for robust training.

Web Application Development

The trained CNN model was integrated into a Flask-based web application. Users can register and log in securely using valid credentials. After authentication, users are allowed to upload facial images through the application interface. Uploaded files are validated for supported formats before processing.

The image is then resized and normalized during runtime preprocessing. The processed image is passed to the trained CNN model, which predicts whether the child belongs to the autistic or non-autistic class. The final result is displayed to the user along with a confidence score. Prediction records and uploaded image details are stored in the database for maintaining user history and future reference.

Technologies Used

Python was used as the primary programming language due to its strong support for machine learning and web development. Flask served as the backend web framework for routing and model integration. Jupyter Notebook was used for model experimentation and training. Anaconda helped manage dependencies and package environments. SQLite was used as the relational database for storing user and prediction records. HTML provided webpage structure, CSS improved visual styling, and JavaScript enabled interactive frontend behavior.

Pseudocode

The implementation workflow can be summarized through the following pseudocode:

Start the system. Load image dataset from storage. Assign labels to images based on folder names.

Preprocess images by resizing and normalization. Split dataset into training and validation sets. Build CNN model with convolution, pooling, dropout, and dense layers. Compile the model using Adam optimizer and binary cross entropy loss. Train the model for defined epochs. Evaluate model performance using accuracy and other metrics. Save trained model weights.

Initialize Flask application. Register and authenticate users. Accept uploaded facial image from user. Validate file format. Preprocess image and send it to the trained CNN model. Generate prediction and confidence score. Display result to user. Store prediction history in database. End process.

Testing

Testing is a critical phase used to verify that the developed application functions according to requirements and remains free from defects. It improves product quality, enhances reliability, and reduces deployment risks. In AI-based healthcare systems, testing is particularly important because inaccurate predictions may affect user trust and decision-making.

The key benefits of testing include cost reduction, improved customer satisfaction, stronger security, and better software quality.

Dimensions of Testing

Testing was performed across multiple dimensions including database testing, API testing, and user interface testing. It was also conducted at different scales such as unit testing, module testing, integration testing, and complete scenario testing. Functional, performance, and security aspects were considered during evaluation. Both manual and automated testing approaches were used where appropriate.

Stages of Testing

Unit testing focused on validating individual modules such as registration, login authentication, image preprocessing, and prediction functions. Each module was tested independently to verify correct outputs.

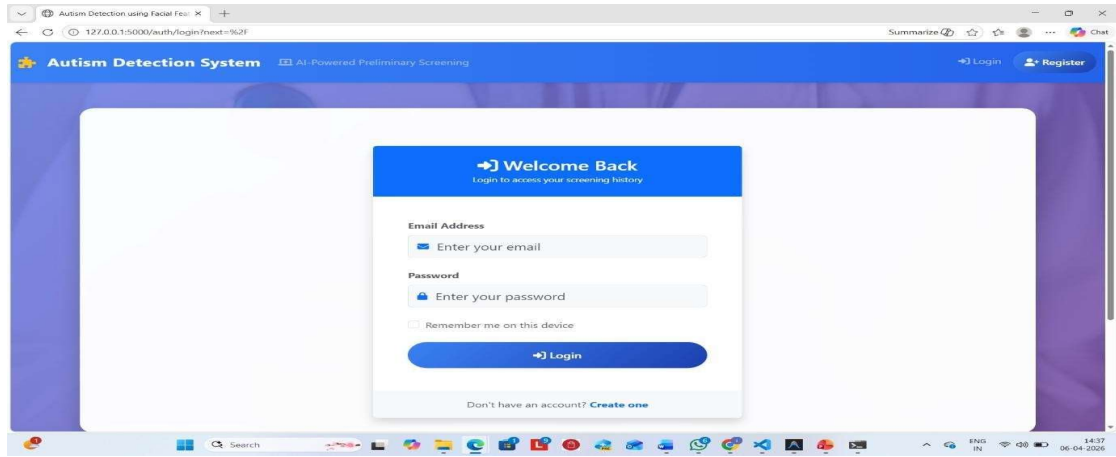
Integration Testing

Integration testing ensured that multiple modules interacted properly. For example, uploaded images had to pass correctly from frontend upload forms to preprocessing modules and then to the CNN model.

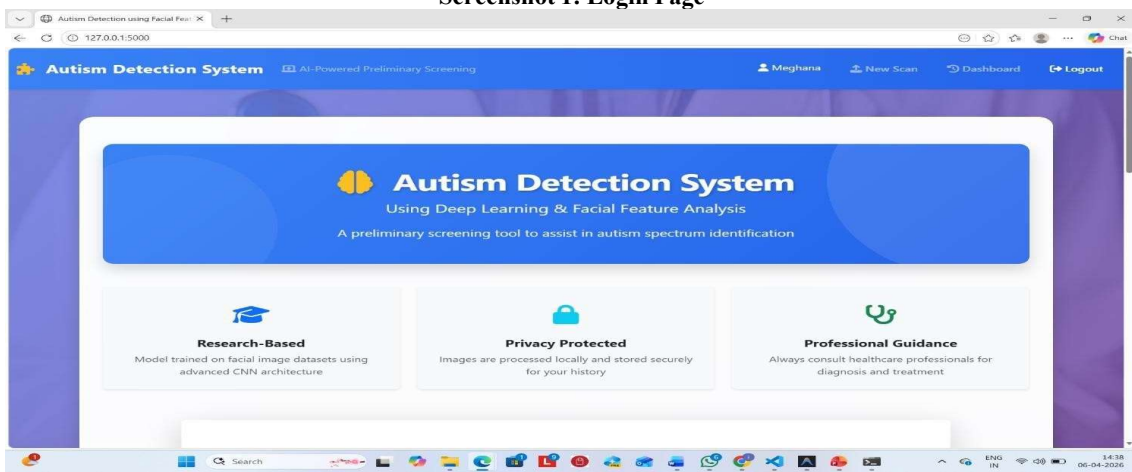
System Testing

System testing evaluated the complete application as an integrated system. It verified whether all features including authentication, prediction, database storage, and result display worked correctly.

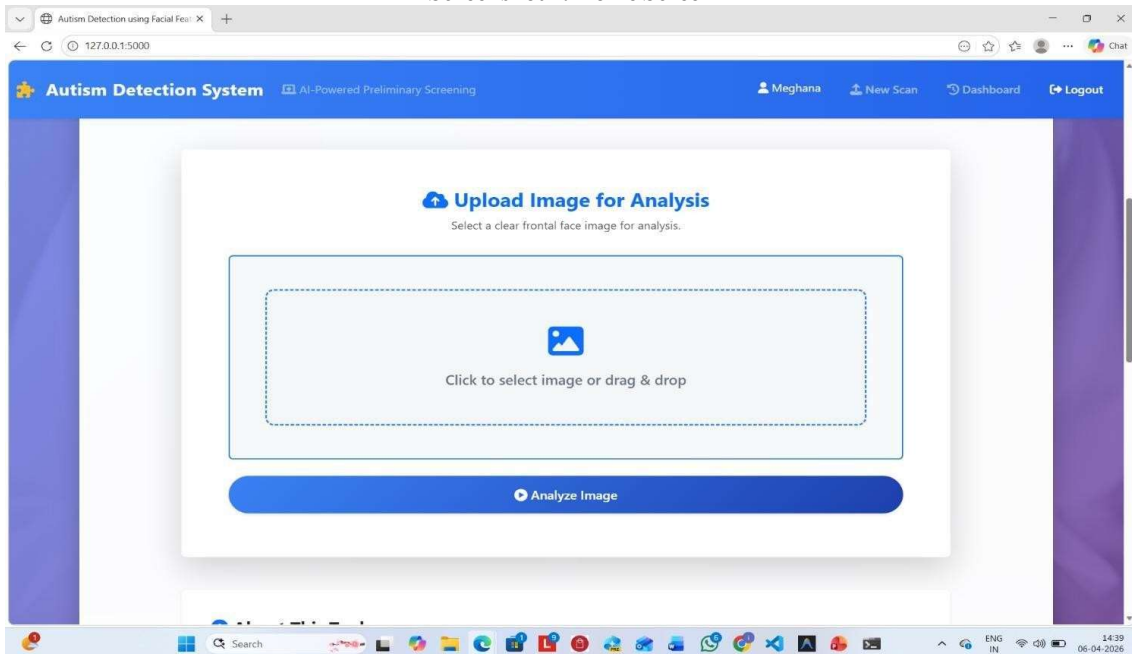
Screenshots



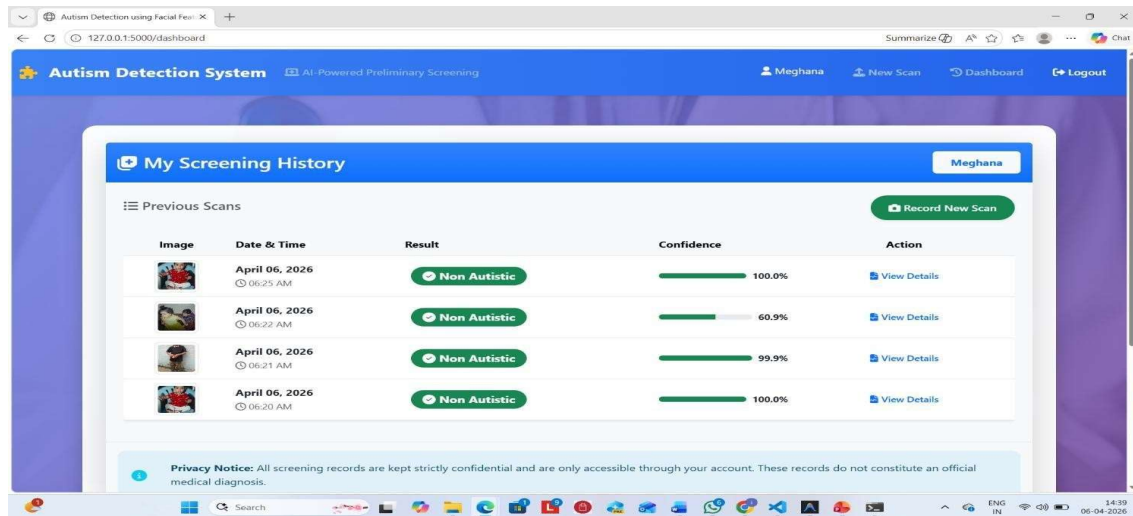
Screenshot 1: Login Page



Screenshot 2: Home Screen



Screenshot 3: Upload Image for Analysis



4 Screening History

Conclusion

This study demonstrates that static facial image analysis combined with deep neural networks can provide an effective approach for the early identification of Autism Spectrum Disorder in children. The proposed system reduces limitations associated with manual diagnosis by offering rapid, consistent, and accessible screening support. Although it is not intended to replace medical professionals, it can significantly assist clinicians and parents in recognizing early warning signs of autism.

Future Scope

Future improvements may include the use of larger and more diverse datasets to enhance model reliability across different populations. Mobile application deployment can increase accessibility for remote users. The integration of behavioural, speech, and eye-tracking data with facial analysis may further improve prediction accuracy. Explainable AI techniques can also be incorporated to make model decisions more transparent and trustworthy.

References

- [1] Centers for Disease Control and Prevention, "Data and Statistics on Autism Spectrum Disorder," Accessed: Mar. 2, 2022. [Online]. Available: <https://www.cdc.gov/ncbddd/autism/data.html>.
- [2] J. Baio, "Prevalence of autism spectrum disorder among children aged 8 years—Autism and Developmental Disabilities Monitoring Network, 11 sites, United States, 2010," *MMWR Surveillance Summaries*, vol. 63, no. 2, pp. 1–21, Mar. 2014.
- [3] B. Mozolic-Staunton, M. Donnelly, J. Yoxall, and J. Barbaro, "Early detection for better outcomes: Universal developmental surveillance for autism across health and early childhood education

settings," *Research in Autism Spectrum Disorders*, vol. 71, Art. no. 101496, Mar. 2020.

- [4] R. C. Sheldrick, S. Markovitz, D. Garfinkel, A. S. Carter, and E. C. Perrin, "Comparative accuracy of developmental screening questionnaires," *JAMA Pediatrics*, vol. 174, no. 4, pp. 366–374, 2020.
- [5] D. L. Robins, K. Casagrande, M. Barton, C.-M.-A. Chen, T. Dumont-Mathieu, and D. Fein, "Validation of the modified checklist for autism in toddlers, revised with follow-up (M-CHAT-R/F)," *Pediatrics*, vol. 133, no. 1, pp. 37–45, Jan. 2014.
- [6] C. Lord, M. Rutter, and 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*, vol. 24, no. 5, pp. 659–685, Oct. 1994.
- [7] K. Pierce, V. H. Gazestani, E. Bacon, C. C. Barnes, D. Cha, S. Nalabolu, L. Lopez, A. Moore, S. Pence-Stophaeros, and E. Courchesne, "Evaluation of the diagnostic stability of the early autism spectrum disorder phenotype in the general population starting at 12 months," *JAMA Pediatrics*, vol. 173, no. 6, pp. 578–587, 2019.
- [8] M. J. Uddin, M. M. Ahamad, S. Aktar, N. K. Alotaibi, S. A. Alyami, M. A. Kabir, and M. A. Moni, "An integrated statistical and clinically applicable machine learning framework for the detection of autism spectrum disorder," Apr. 2023.
- [9] H. Manohar, P. Kandasamy, V. Chandrasekaran, and R. P. Rajkumar, "Early diagnosis and intervention for autism spectrum disorder: Need for pediatrician–child psychiatrist liaison," *Indian Journal of Psychological Medicine*, vol. 41, no. 1, pp. 87–90, Jan. 2019.
- [10] H. Shamseddine, S. Otoum, and A. Mourad, "On the feasibility of federated learning for neurodevelopmental disorders: ASD detection use-case," Dec. 2022.