

Early ASD Insight -Autism Spectrum Disorders

Samir Kumar,
Computer Science and Engineering
Dayananda Sagar University
Devarakaggalahalli , Ramanagar,
Bangaluru, 562112, Karnataka, India
eng22cs0435@dsu.edu.in

Srinivas Reddy,
Computer Science and Engineering
Dayananda Sagar University
Devarakaggalahalli , Ramanagar,
Bangaluru, 562112, Karnataka, India
eng22cs0592@dsu.edu.in

Sujal Narayan,
Computer Science and Engineering
Dayananda Sagar University
Devarakaggalahalli , Ramanagar,
Bangaluru, 562112, Karnataka, India
eng22cs0593@dsu.edu.in

Utkarsh Singh,
Computer Science and Engineering
Dayananda Sagar University,
Devarakaggalahalli , Ramanagar,
Bangaluru, 562112, Karnataka,
India eng22cs0488@dsu.edu.in

Santhosh M,
Assistant Professor
Dayananda Sagar University
Devarakaggalahalli , Ramanagar,
Bangaluru, 562112, Karnataka, India
santosh.m-cse@dsu.edu.in

Abstract - Early identification of Autism Spectrum Disorder (ASD) in children is essential for timely intervention and improved developmental outcomes. The worldwide increase in autism cases has created a demand for accurate diagnostic tools that can be used by all people. The researchers of this study use facial image datasets to predict Autism Spectrum Disorder (ASD) in its initial stages. The main of this study is to explore whether distinctive facial patterns and expressions can serve as potential indicators of ASD. Researchers use advanced image processing techniques to extract facial image features which they use to develop machine learning systems that predict autism, autism risk through classification. The study tests the hypothesis that certain facial characteristics may be associated with early signs of ASD and can be detected using automated computational models. The research process includes three main steps which involve acquiring a complete set of children's images, performing systematic image processing to enhance image quality and extract essential features, and using advanced deep learning frameworks for classification. The research uses computer vision combined with machine learning methods to create tools that enable non- invasive autism screening which operates efficiently and is easy to use.

I. INTRODUCTION

Autism Spectrum Disorder (ASD) is a complex neurodevelopmental condition which involves various problems with behavior and communication and social interaction. The process of diagnosing ASD presents difficulties because different people show different symptoms which can also be found in other developmental disorders. The combination of these two factors creates difficulties for the process of finding and diagnosing conditions at an early stage. The traditional diagnostic approach depends on clinicians who observe patient behavior and assess their condition through clinical assessments which lose objectivity when applied to different environments. The research study aims to develop existing diagnostic systems through it's study of facial feature assessment which will help detect autism spectrum disorder. (ASD). Human facial features convey a wealth of subtle

information which includes expressions and structural characteristics that may reflect underlying neuro developmental patterns.

Machine learning research has progressed through the advancement of convolutional neural networks (CNNs) which now enable researchers to study complex visual data and discover hidden patterns that escape human assessment.

The research method uses deep learning methods to analyze facial images which the system uses to identify possible indicators of autism spectrum disorder. The system uses multi- task learning methods to enhance its ability to generalize and adapt which allows the system to understand various ways that autism spectrum disorder presents itself through different people. The method improvement enhances the predictive model strength through development of more powerful prediction capabilities. This research brings important societal benefits which extend beyond its technical achievements. The ability to detect autism spectrum disorder through reliable methods enables healthcare professionals to provide immediate treatment which leads to better developmental progress.

II. RELATED WORK

The authors of [1] proposed a deep learning-based web- based system which helps families and psychiatrists to diagnose Autism Spectrum Disorder (ASD) through facial image analysis. The system used convolutional neural networks (CNNs) which implemented transfer learning using the Flask framework. The system used MobileNet and Xception and InceptionV3 as pretrained models for its classification tasks. The Kaggle dataset contained 3,014 facial images which researchers used to show both autistic and non-autistic children. The MobileNet system achieved 95% accuracy while Xception reached 94% and InceptionV3 achieved approximately 89% accuracy. The study proved that lightweight pretrained CNN models succeed in detecting autism through facial feature analysis.

The researchers of [2] developed a facial attention detection framework which uses two separate methodologies for its operation. The first approach used geometric feature extraction together with the Support Vector Machine classifier while the second approach analyzed time-domain spatial features through two-dimensional representations which were processed by a CNN model. The study tested 46 children who included 20 children with ASD and 26 typically developing children to complete different attention-related tasks. The results showed that SVM-based geometric feature transformation performed better than the CNN-based method. The model demonstrated improved generalization abilities for typically developing children while displaying better performance on low-attention tasks than on high-attention task

In [3] the authors investigated whether face-scanning patterns could be used to distinguish children with ASD from typically developing children. A machine learning algorithm was applied to analyze eye-tracking data collected during a face recognition task. The model was evaluated using accuracy, sensitivity, and specificity metrics, achieving a maximum classification accuracy of 88.51%. Although the results were promising, the study acknowledged limitations in clinical applicability and recommended further development of multi-task and multi-model frameworks for improved early detection.

In [4] a deep learning-based facial analysis model was developed to classify children as either autistic or healthy. The model utilized MobileNet along with additional dense layers for feature extraction and classification. Trained and tested on 3,014 facial images, the system achieved an accuracy of 94.6%. The findings suggested that facial image analysis can serve as a cost-effective and efficient method for early autism screening.

In [5] the researchers examined emotion recognition abilities in individuals with ASD using the Films Expression Task. The study involved 46 people with ASD and 52 age- and IQ-matched typically developing individuals. Results revealed significant differences in recognition accuracy and reaction times, indicating notable deficits in facial expression processing among individuals with ASD.

However, a small subgroup of participants with ASD demonstrated performance comparable to typically developing individuals, highlighting variability within the spectrum

A large-scale meta-analysis was conducted in [6] to study face identity processing deficits which occur in individuals with ASD. The research study analyzed 112 research papers which included 5,390 participants and showed that individuals with ASD demonstrated significant impairments during both face recognition and face discrimination tests. The research results showed that face identity processing difficulties constitute an essential feature of autism spectrum disorder which stays constant across different age groups and gender and testing methods.

The research team in [7] used machine learning methods to create a system which separates children with autism spectrum disorder according to their facial scanning behavior. The model reached its highest performance level when it used eye-tracking data from autistic children and typically developing children, achieving an accuracy rate of 88.51%. The research developed an automatic classification system while stressing the importance of testing it with more extensive data sets for validation purposes

The authors in [8] used five pretrained CNN models which include MobileNet and Xception and EfficientNetB0 and EfficientNetB1 and EfficientNetB2 to extract features before using a deep neural network to perform binary classification. Xception performed better than all other models because it achieved an AUC score of 96.63% combined with 88.46% sensitivity and 88% negative predictive value. The research showed that researchers can successfully obtain unique facial characteristics which are typical for ASD from unchanging facial photos.

In [9] researchers developed a real-time emotion recognition system specifically designed for children with autism through their improved deep learning method. The system detected six facial emotions and achieved a reported accuracy of 99.99% on a cleaned dataset. The authors explained that their study encountered limitations because of their small dataset size which they suggested should be investigated through studies that utilize larger datasets from actual environments

In [10] researchers created a real-time facial emotion recognition system which helps children with ASD learn about emotions. The system included stages for face detection, feature extraction, and classification, achieving an average recognition rate of 85.97%. The study focused primarily on improving empathy and emotional understanding in children with autism.

The research study has demonstrated important progress through its earlier work which still shows multiple restrictions that include existing data set limitations and inability to apply results to different groups and the study's missing elements of multi-task learning together with ensemble methods. Existing methods currently either extract facial features or analyze human behavior but they do not use multiple facial features together with deep learning techniques to create better results.

III. METHODOLOGY

The proposed system for the Autism Spectrum Disorder (ASD) classification project introduces a comprehensive approach that integrates advanced facial attribute analysis with deep learning, specifically leveraging convolutional neural networks (CNNs). The system seeks to improve the accuracy of ASD diagnosis by incorporating an ensemble technique, combining multiple CNN architectures, such as Densenet 121, Mobile Net, Xception, and InceptionV3, to create a robust model.

The workflow of the system involves the extraction of relevant facial attributes, including expressions, action units, arousal, and valence, from a diverse dataset of images. These attributes are then fed into a multi-task learning framework within CNNs, allowing the model to simultaneously analyze various facial features crucial for ASD classification.

To further boost accuracy, the ensemble technique is applied, where individual CNN models are trained using different subsets of the dataset or with varying hyperparameters. The predictions generated by each model are combined, harnessing the collective intelligence of diverse models and mitigating the risk of overfitting.

Ethical considerations are embedded in the system, with a focus on privacy safeguards for sensitive facial data. The societal impact is considered, aiming to improve the early diagnosis of ASD, potentially leading to more effective interventions and reduced healthcare costs. Environmental concerns related to energy consumption during training and data storage are addressed by optimizing the computational resources.

Validation and testing processes involve diverse datasets, ensuring the generalizability of the proposed system across different demographics. The documentation is maintained rigorously, allowing for transparency, reproducibility, and continuous improvement. Overall, the proposed system represents an innovative and ethically conscious approach to ASD diagnosis, utilizing the synergy of facial attribute analysis, deep learning, and ensemble techniques to achieve higher accuracy and societal impact.

A. Covid-19 Classification using Dense Net 121 Model:

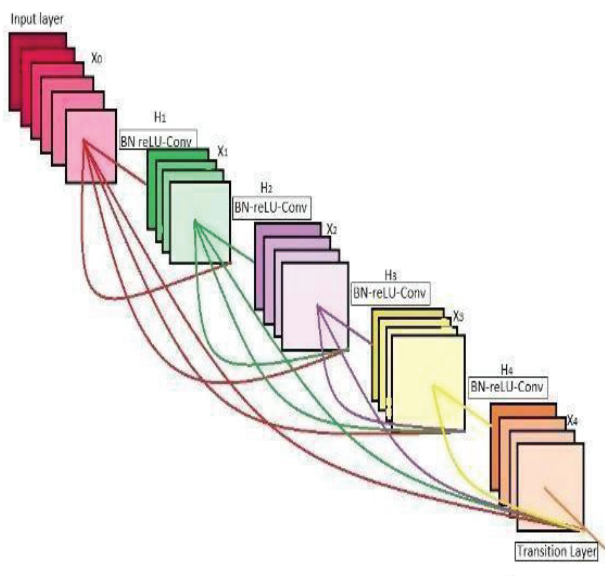


Figure 1: DenseNet121 Model

The input layer of the proposed model receives medical images obtained from multiple imaging modalities, including preprocessed and enhanced X-ray, computed tomography (CT), and ultrasound scans. The network needs these images to complete its operation which requires the images to go through standard preprocessing steps that include noise reduction and normalization and contrast enhancement to achieve better image quality and modal consistency.

Convolutional Layers- The Convolutional Neural Network (CNN) architecture depends on the convolutional layer as its main element that extracts features from input data. The layer executes convolution operations which are followed by the implementation of activation functions that introduce nonlinearity into the system. A small matrix known as a kernel or filter moves across the input image during the convolution process. The filter values get multiplied with corresponding input values for each spatial location, and the outputs of this operation create one final output value. The system executes this process throughout the entire input, which results in creating output matrices that scientists call feature maps. The feature maps of the system extract essential local patterns, which include edges and textures and shapes and all other features that help in distinguishing different objects. The stacking of multiple convolutional layers enables the network to develop complex abstract input data representations which become vital for achieving correct classification and detection results.

Pooling Layers- The feature maps that convolutional layers create need their spatial dimensions to be reduced through pooling layers. The main goal of pooling operations is to decrease processing requirements while keeping essential data. The system processes each feature map by splitting it into multiple distinct sections that do not overlap with each other. A summary stat

$$MP = \text{Floor} \left(\frac{I_x - P}{s} + 1 \right) \quad (1)$$

The dense layer of a Convolutional Neural Network (CNN) implements a linear transformation which processes input features from the prior layer of either a pooling layer or another dense layer. This transformation process begins with matrix multiplication of the input vector and weighted matrix, followed by adding the bias term. Every neuron in a dense layer maintains active connections to all neurons present in the previous layer, which leads to its identification as a "fully connected" network. This complete connectivity of the network enables it to learn advanced mapping functions which can handle intricate patterns between extracted features and actual results.

The dense layer functions as a processing unit that transforms high-level data obtained through convolutional and pooling methods into a condensed yet distinct form of information. The final classification or regression task of the CNN architecture is executed by the dense layer, which operates as the concluding element of the system to perform tasks such as image categorization and segmentation and object detection. The dense layer output converts to final predictions through an activation function which uses either sigmoid for binary classification or Softmax for multi-class classification. The dense layer serves a vital function which enables the system to transform learned feature representations into valuable output.

Dropout Layer Dropout serves as a regularization method which is frequently used in CNNs to reduce the risk of overfitting. When a model experiences overfitting, it becomes excessively customized to the training data, which prevents it from performing well on new, untrained data. The dropout layer deactivates a predetermined percentage of neurons during training by turning their outputs to zero throughout the forward pass process

The iteration does not use these neurons because they stop functioning during both forward propagation and backpropagation. The network achieves its needed functionality because the stochastic neuron removal process prevents it from depending on particular features and specific neurons. Dropout forces the model to use different routes for making predictions, which helps the system learn stronger and more universal feature representations. The network creates extra representation systems, which allow it to maintain performance when processing unfamiliar information. The dropout rate hyperparameter controls how many neurons become inactive during the training process. The training experiments use dropout rates that typically fall between 0.2 and 0.5, but the ideal rate depends on the dataset size and network complexity and task requirements.

$$\frac{\partial}{\partial s_p} \left(-\log \left(\frac{e^{s_p}}{\sum_j e^j} \right) \right) = \left(\frac{e^{s_p}}{\sum_j e^j} - 1 \right) \quad (2)$$

$$\frac{\partial}{\partial s_n} \left(-\log \left(\frac{e^{s_p}}{\sum_j e^{s_j}} \right) \right) = \left(\frac{e^{s_n}}{\sum_j e^{s_j}} \right) \quad (3)$$

Activation Function In a Convolutional Neural Network (CNN), an activation function processes the output of each neuron, introducing non-linearity to the model. The primary purpose of this non-linear transformation is to empower the network to model complex mappings from inputs to outputs. A CNN requires activation functions because they establish its nonlinear behavior which enables the system to understand complex patterns found in actual data. CNN architectures make use of multiple activation functions which are frequently implemented. The Rectified Linear Unit (ReLU) function together with Sigmoid and Hyperbolic Tangent (Tanh) functions are among the most commonly used activation functions. The ReLU activation function is the most frequently used in modern CNNs. Negative input values become zero, whereas positive ones remain the same. ReLU demonstrates efficient computation abilities while it solves the vanishing gradient problem which frequently appears in deep neural network systems. Its simplicity and effectiveness make it a preferred choice for hidden layers. Sigmoid maps inputs to outputs between 0 and 1 which makes it suitable for binary classification problems that require outputs to show probabilities. The Sigmoid function in deep networks will experience the vanishing gradient problem which leads to decreased training speed. The Tanh activation function maps input values to a range between -1 and 1. The zero-centered nature of this function enables better convergence performance than the Sigmoid function in certain situations. Tanh is useful when modeling data that spans both positive and negative values.

The SoftMax activation function serves as the standard output layer for multi-class classification tasks. SoftMax converts raw output scores (logits) into probability distributions, where the sum of all class probabilities equals one. The system enables multiple class probability assessments while choosing the category with highest probability. Activation functions hold crucial importance for CNNs because they establish non-linear characteristics which enable networks to learn intricate data patterns that resemble real-world phenomena. Among them, ReLU is most commonly employed in hidden layers owing to fast computation and strong performance to alleviate gradient-related issues, while Sigmoid and SoftMax are often used in output layers depending on the classification task.

The SoftMax function serves as the verification method for multi-class classification tasks. The task of multi-class classification requires operators to select a single category from multiple available options to classify an input sample. A CNN uses its training process to learn how to identify different object categories which include specific animal breeds and various types of vehicles.

The SoftMax activation function serves as the standard activation function which developers use to build their last output layer. The system provides output probabilities which remain bounded between 0 and 1 while their total value equals 1. The system predicts classes by selecting the category which has the highest probability value. The SoftMax function allows users to understand model results as probability distributions which make it ideal for solving multi-class classification tasks.

Table 1: DenseNet121 Parameter

block_16_expand_BN (BatchNormal)	(None, 7, 7, 960)	3840	block_16_expand[0][0]
block_16_expand_relu (ReLU)	(None, 7, 7, 960)	0	block_16_expand_BN[0][0]
block_16_depthwise (DepthwiseCo)	(None, 7, 7, 960)	8640	block_16_expand_relu[0][0]
block_16_depthwise_BN (BatchNor)	(None, 7, 7, 960)	3840	block_16_depthwise[0][0]
block_16_depthwise_relu (ReLU)	(None, 7, 7, 960)	0	block_16_depthwise_BN[0][0]
block_16_project (Conv2D)	(None, 7, 7, 320)	307200	block_16_depthwise_relu[0][0]
block_16_project_BN (BatchNorma)	(None, 7, 7, 320)	1280	block_16_project[0][0]
Conv_1 (Conv2D)	(None, 7, 7, 1280)	409600	block_16_project_BN[0][0]
Conv_1_bn (BatchNormalization)	(None, 7, 7, 1280)	5120	Conv_1[0][0]
out_relu (ReLU)	(None, 7, 7, 1280)	0	Conv_1_bn[0][0]
global_average_pooling2d (Globo)	(None, 1280)	0	out_relu[0][0]
Logits (Dense)	(None, 1000)	1281000	global_average_pooling2d[0][0]
=====			
Total params: 3,538,984			
Trainable params: 3,504,872			
Non-trainable params: 34,112			

Optimization Algorithms in CNN Training The optimization algorithm determines how CNN training works because it controls the process which updates network parameters through weight and bias modifications to achieve loss function reduction. The loss function calculates the discrepancy between predicted outputs and actual target values, and the optimization process targets to find the parameter configuration that produces the lowest possible loss. The training process uses gradient-based techniques which operate through iterative cycles. The system calculates loss function gradients which measure parameter changes during each backpropagation cycle, then updates parameters based on those calculations. Several optimization techniques are commonly used in CNN training, including: Stochastic Gradient Descent (SGD) – adjusts parameters using gradients derived from mini-batches of training data. The method provides a basic solution which functions well, yet it has a slow speed for reaching complete convergence. Adaptive Moment Estimation (Adam).

The system achieves faster convergence with improved stability through its combination of momentum and adaptive learning rate techniques. Root Mean Square Propagation (RMSprop) – The system modifies learning rates according to current gradient strengths, which results in better performance during periods of changing conditions. Each optimization algorithm has distinct strengths and limitations, and the choice of optimizer can significantly influence convergence speed, stability, and overall model performance. The optimization process executes as an iterative framework which improves generalization ability through continuous adjustments of network parameters until prediction errors reach their lowest point.

The Adam Optimization Algorithm Every optimization algorithm provides specific benefits, while it creates particular drawbacks. The choice of optimizer determines how quickly a

$$Accuracy = \left(\frac{TP + TN}{TP + TN + FP + FN} \right) \quad (4)$$

trained neural network achieves its performance target through

its learning process. The process of optimization involves multiple iterations, which update network parameters like weights and biases till the loss function obtains its minimum value and prediction accuracy shows improvement. The Adaptive Moment Estimation algorithm functions as the most popular method among deep learning optimization algorithms, which researchers use to train Convolutional Neural Networks. Adam presents its users with the advantages obtained from Stochastic Gradient Descent and Root Mean Square Propagation. The system determines learning rates for each parameter through its dual estimation process, which includes measuring past gradient averages and calculating squared gradient averages. Adam achieves faster training times through its capacity to change learning rates throughout the training process. The method demonstrates strong performance when dealing with both high-frequency data and low-frequency data, as well as through its application to intricate network designs. Deep learning systems demonstrate fast convergence rates because of their ability to process large-scale data sets. The system operates efficiently across various tasks because it requires minimal hyperparameter adjustments through manual tuning. The optimizer exists as an effective tool for CNN training, which provides both fast performance and dependable results.

Evaluation Metrics A CNN model needs suitable quantitative evaluation metrics to assess its performance. The metrics enable assessment of model classification performance and enable comparison of different modeling methods. Prediction results are typically summarized through the use of a confusion matrix. The system displays its performance through two types of classification outcomes which include correct classifications and incorrect classifications. The evaluation metrics which stem from the confusion matrix provide multiple assessment options. Accuracy reflects the ratio of right predictions to the total samples tested. The dataset imbalanced nature leads to complications which make it difficult to understand accuracy results even though it remains an understandable metric.

$$PR = \left(\frac{TP}{TP + FP} \right) \quad (5)$$

Precision reflects the reliability of positive predictions. The measurement shows what percentage of positive predictions turn out to be actual positive results. Recall, or sensitivity, evaluates how well the model identifies actual positives. The system shows its capacity to discover important information through its performance. The F1-score metric enables equal evaluation through its combination of precision and recall into one measurement. The method proves beneficial for handling situations that involve unequal data distribution. The Area Under the Curve (AUC) metric tests a model's capacity to differentiate between various classes at multiple classification levels. A higher AUC value indicates better discriminatory capability. Multiple evaluation metrics enable better understanding of how a model performs during assessments. The combined implementation of accuracy and other metrics helps to reveal deeper examination results for class imbalance situations through their implementation of precision, recall, F1-score.

The confusion matrix shows True Positive (TP) when the model successfully identifies a normal image and the actual label also shows normal status. The model successfully detects an abnormal image when it produces True Negative (TN) output which matches the actual abnormal designation. The model treats an image as normal when it produces False Positive (FP) results because the actual condition of the image is abnormal. The False Negative (FN) error occurs when the model shows an image as abnormal while the actual label shows it as normal. The four different results serve as fundamental elements which researchers use to assess how well a system performs classification tasks. Precision calculates the fraction of positive predictions which the system makes to determine actual positive cases. The system assesses the accuracy of its positive predictions through this metric. A model with high precision produces fewer false positive errors. Precision becomes vital for fields which deal with situations where false positive results create severe negative outcomes. The model uses recall which people refer to as sensitivity to test its performance in identifying actual positive cases. The measurement shows what percentage of true positive instances the system successfully identified from all positive instances. The model shows high recall because it generates fewer false negative results. Recall functions as a crucial requirement in situations where overlooking a positive case would create expensive or hazardous results.

AUC measures by and large execution and assesses it over all edges. AUC ranges in esteem from 0 to 1. A show with 100% incorrect forecasts will have an AUC value of 0.0, while a show with 100% adjust expectations will have an AUC value of 1.0. One of the often-employed measurements, for the most part for double classification. A classifier's AUC measures probability.

A confusion matrix tabulates actual versus predicted labels to gauge classifier effectiveness. It offers a visual depiction of the classifier's performance and enables comprehension of the kinds of errors the classifier is committing. Four separate values comprise the confusion matrix:

$$TPR = \frac{TP}{Actual\ Positive} = \frac{TP}{TP + FN}$$

$$FNR = \frac{FN}{Actual\ Positive} = \frac{FN}{TP + FN}$$

$$TNR = \frac{TN}{Actual\ Negative} = \frac{TN}{TN + FP}$$

$$FPR = \frac{FP}{Actual\ Negative} = \frac{FP}{TN + FP}$$

- Instances where the classifier correctly predicted the positive class are known as true positives (TPR).
- When the classifier predicted a positive class when a negative class was actually present, this is known as a false positive (FPR).
- Situations where the classifier correctly identified the negative class are known as True Negatives (FNR).
- When the classifier correctly identified the negative class, the scenario is known as a True Negative (TNR).

The performance of models can be assessed through the combined examination of precision and recall.

Precision measures the rate of incorrect positive results while recall measures the rate of incorrect negative results. The two values need to reach their maximum value of 100 percent according to ideal conditions

$$REC = \left(\frac{TP}{TP + FN} \right) \quad (6)$$

Precision improvement leads to decreased FP errors, whereas recall improvement results in decreased FN errors. The process of enhancing one aspect of performance may create adverse effects on another aspect which results in a performance trade-off. The F1-score serves as a comprehensive metric that integrates precision and recall into a single balanced value by computing their harmonic mean, ensuring both metrics receive equal importance in evaluation. The F1-score proves beneficial when handling datasets with imbalanced classes and when both false positive and false negative errors need consideration. The system evaluates classification results through precision and recall which provides an unbiased assessment of classification results.

$$F1 - Score = \left(\frac{2 * PR * REC}{PR + REC} \right) \quad (7)$$

IV. CONCLUSION

In conclusion, the ASD classification project stands as a noteworthy achievement in leveraging advanced facial attribute analysis and an ensemble of CNN architectures for Autism Spectrum Disorder diagnosis. With a commendable 91% validation accuracy achieved over 45

training epochs, the model adeptly learns nuanced facial features, including expressions, action units, arousal, and valence. The dynamic progression of accuracy, coupled with a balanced data loss trade-off, underscores the model's adaptability to diverse patterns, contributing to its reliability in real-world applications. The streamlined testing model, designed for user-friendly image path input, enhances accessibility, providing binary predictions for efficient and practical ASD diagnosis. Overall, this project represents a significant step forward in the integration of advanced technologies for accurate and accessible ASD classification.

In practical terms, the testing model offers a straightforward tool for users to input image path, receiving prompt binary predictions categorizing individuals as either autistic or non-autistic. Beyond its high validation accuracy, the project's emphasis on user-friendly implementation ensures its applicability in diverse settings, contributing to the ongoing evolution of ASD diagnostic methodologies.

REFERENCES

- [1] Facial features detection system to identify children with autism spectrum disorder using deep learning models. Available: <https://www.researchgate.net/publication/359710714>
- [2] Face-based attention recognition model for children with autism spectrum disorder. Available: <https://www.researchgate.net/publication/353278709>
- [3] Identifying children with autism spectrum disorder based on their face processing abnormality: A machine learning framework. Available: <https://www.researchgate.net/publication/299585544>
- [4] Deep learning for autism diagnosis and facial analysis in children. Available: <https://www.researchgate.net/publication/357998269>
- [5] Facial expression recognition as a candidate marker for autism spectrum disorder: How frequent and severe are deficits. Available: <https://www.researchgate.net/publication/322811320>
- [6] A quantitative meta-analysis of face recognition deficits in autism: 40 years of research. Available: <https://www.researchgate.net/publication/346416614>
- [7] Identifying children with autism spectrum disorder via machine learning-based behavior analysis. Available: <https://kithub.cmu.edu/articles/thesis/22221010>
- [8] Identification of autism in children using static facial features and deep neural networks. Available: <https://www.researchgate.net/publication/357778747>
- [9] Neural Computing and Applications. Available: <https://www.researchgate.net/journal/Neural-Computing>
- [10] AutisMitr: Emotion recognition assistive tool for autistic children. Available: <https://www.researchgate.net/publication/343258823>