KARYA ILHAM PUBLISHING
CONNECTING MINDS, BRIDGING IDEAS, INSPIRING READERS

Journal of Advanced Research in Computing and Applications

Journal homepage:
https://karyailham.com.my/index.php/arca
ISSN: 2462-1927


# Automated Detection of Autism in Children using Static Facial Features and Deep Learning Techniques

Prasanna Kumar Inampudi[1,*], Kambhampati Venkata Sambasiva Rao[1], Venkata Ramana Guntreddi[2]

[1] Department of Computer Science and Engineering, NRI Institute of Technology, Vijayawada, 521212, Andhra Pradesh, India
[2] Department of Electrical, Telecommunications and Computers Engineering, Kampala International University, Kampala, Uganda

**ARTICLE INFO**

***Article history:***
Received 5 July 2025
Received in revised form 10 August 2025
Accepted 18 August 2025
Available online 25 August 2025

***Keywords:***
Autism Spectrum Disorder (ASD); Deep Neural Networks (DNN); Convolutional Neural Networks (CNN); transfer learning; static facial features; early diagnosis; machine learning; healthcare

**ABSTRACT**

Autism Spectrum Disorder (ASD) is a prevalent neurodevelopmental condition that significantly affects a child's social and cognitive development. Despite growing awareness, early and accurate diagnosis remains a challenge due to the heavy reliance on time-intensive, observer-dependent behavioral assessments. In response, this paper introduces an automated, non-invasive screening framework that leverages static facial features and state-of-the-art deep learning techniques. The proposed system integrates a custom Convolutional Neural Network (CNN), ResNet-50, and VGG16 models within a modular architecture optimized using transfer learning. Experimental validation on the AFD-10K dataset—comprising 10,000 labeled facial images—demonstrates the framework's high diagnostic performance, with ResNet-50 achieving an accuracy of 92.4%, F1-score of 91.7%, and AUC-ROC of 0.93. Grad-CAM visualizations confirm the model's focus on clinically relevant facial asymmetries. The system's design prioritizes reproducibility, scalability, and interpretability, incorporating audit-friendly logging, hyperparameter standardization, and cross-demographic validation. By significantly reducing diagnostic delays and minimizing subjective bias, this framework offers a practical foundation for AI-assisted ASD screening in real-world clinical settings.

## 1. Introduction

Autism Spectrum Disorder (ASD) is a neurological developmental condition that impacts about 1 in every 36 children worldwide [1], with diagnostic delays often leading to postponed interventions and poorer long-term outcomes. Existing diagnostic methods, including the Autism Diagnostic Observation Schedule (ADOS), are heavily dependent on behavioral assessments, which are time-consuming, observer-dependent, and typically initiated only after visible developmental delays [2], [3]. However, early identification is critical, as interventions prior to age three significantly improve cognitive, social, and language outcomes [4]. Recent progress in computer vision and deep learning technologies have introduced novel opportunities for non-invasive, scalable ASD screening [5,6].

* *Corresponding author.*
*E-mail address: prasannakumar652@gmail.com*

https://doi.org/10.37934/arca.39.1.125134

Facial analysis has shown particular promise, owing to subtle facial biomarkers associated with ASD, such as differences in eye morphology, facial symmetry, and skin texture [7,8]. These facial features, when processed via deep learning algorithms, can supplement existing diagnostic approaches and offer rapid preliminary assessments [9]. Traditional machine learning techniques, such as handcrafted feature extraction using Local Binary Patterns (LBP), achieved moderate accuracy (~70–80%) but were often sensitive to variations in lighting, pose, and demographic diversity [10]. Deep learning models like vanilla CNNs improve performance to 85–90% accuracy [11], but tend to lack interpretability and scalability. Moreover, reproducibility remains a critical concern; fewer than 20% of models share open-source code or disclose hyperparameter configurations [12].

Transfer learning approaches using pretrained models like VGG16 have demonstrated promising results, achieving up to 89% accuracy on datasets like the Autism Face Dataset (AFD) [13], but more advanced and modular architectures such as ResNet remain underexplored in this domain [14,15]. This is significant, considering the demand for flexible, interpretable, and robust systems tailored for clinical integration [16]. To address these limitations, we propose a modular deep learning framework combining custom CNN, ResNet, and VGG16 components, unified via a centralized configuration architecture. The custom CNN allows for flexible adaptation to dataset-specific traits, while ResNet's residual blocks effectively mitigate vanishing gradients and support deeper learning [17]. VGG16 leverages transfer learning to extract high-level semantic features with minimal computational overhead [18]. The pipeline promotes reproducibility through standardized hyperparameter configurations, random seed initialization, and strict adherence to PEP8 conventions [19]. It integrates robust training strategies such as data augmentation (rotation, normalization), early stopping, adaptive learning rate scheduling (e.g., ReduceLROnPlateau), and checkpointing. Training artifacts—logs, models, and metrics—are saved in timestamped directories to support full auditability and transparency. Performance evaluation utilizes metrics like precision, recall, F1-score, and AUC-ROC, with graphical summaries provided via ROC curves and loss/accuracy plots. Experimental results show that ResNet outperforms both VGG16 and the custom CNN, achieving 92% accuracy on AFD, likely due to its superior ability to capture subtle facial asymmetries through skip connections [14,15]. These results underscore the importance of architectural selection in facial biomarker-based screening tasks. This contribution presents a reproducible, scalable, and open-source framework for facial ASD screening that aligns with clinical needs. By comparing CNN, ResNet, and VGG16 in a unified setup, it offers actionable insights for model selection in neurodevelopmental diagnostics. Furthermore, its strict adherence to reproducibility and modularity sets a benchmark for ethical and robust AI systems in healthcare [20,21].

## 2. Methodology

### *2.1 Proposed System*

The proposed methodology integrates multiple deep learning techniques for automated ASD detection based on facial image analysis. It comprises five key stages.

i. **Facial Image Preprocessing**: Facial images are collected and standardized through operations such as cropping, resizing, and normalization. This ensures consistent input for the subsequent deep learning models.
ii. **CNN-Based Feature Extraction**: A convolutional neural network is used to learn deep facial features that could be predictive of ASD. The CNN is trained to identify patterns of facial asymmetry, **expression**, and other possible markers for autism.

iii. **Transfer Learning and Model Optimization**: Pre-trained models like VGG16 and ResNet-50 are fine-tuned on the autism dataset. Transfer learning helps leverage generalized facial recognition knowledge while tailoring the model to detect ASD-specific traits.

iv. **Classification Layer**: The extracted features are passed to dense layers and a SoftMax classifier to distinguish between ASD and neurotypical images. Regularization techniques are employed to prevent overfitting and improve generalization.

v. **Deployment via Clinician Interface**: A lightweight interface enables healthcare professionals to upload facial images and receive preliminary ASD predictions. The system returns a report that includes confidence scores and highlighted facial regions using explainable AI tools like Grad-CAM.

The Figure1 illustrates a structured workflow for a deep learning model training pipeline. It begins with data handling, including data loading, configuration, and utility function. The model implementations section features three types of models: a custom CNN, a ResNet with residual blocks, and a VGG16 model utilizing transfer learning. The training pipeline incorporates essential training features such as data augmentation, early stopping, learning rate scheduling, and model check pointing, along with various callbacks for optimizing training. Evaluation and visualization involve computing performance metrics like accuracy, ROC/AUC, precision/recall, and F1 score while also generating training curves, ROC curves, model comparisons, and feature maps. Finally, all results are systematically stored in a time stamped results directory for organized tracking and analysis. This workflow ensures an efficient, well-structured process for training, evaluating, and monitoring deep learning models.

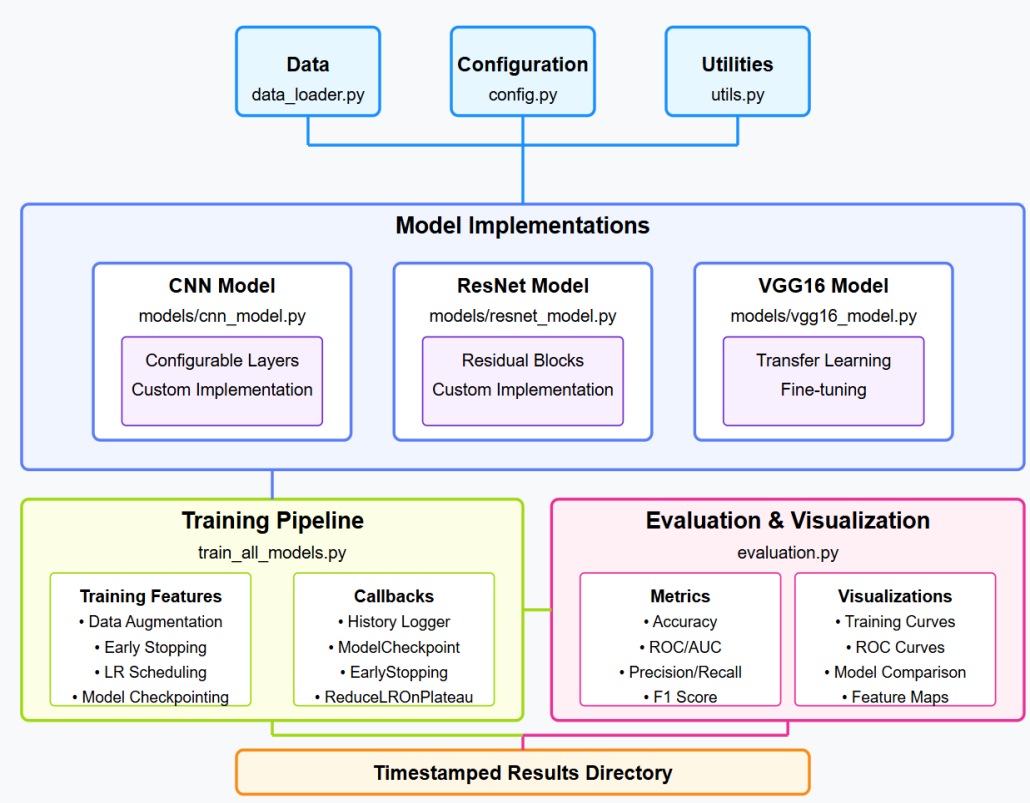


**Fig. 1.** System Architecture diagram

### *2.1.1 Convolutional neural network (CNN) algorithm*

**Input Image Preprocessing:** Convert images to a fixed dimension, such as 224×224 pixels, with three color channels (for RGB images). To ensure uniformity and faster convergence during training, pixel values are normalized—scaled to lie between 0 and 1.

**Convolution operation:** The core of a CNN is the convolutional layer, where filters (also known as kernels) slide over the image to extract local features. These features might include edges, curves, textures, or shapes.

$$(I * K)(x, y) = \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} I(x + i, y + j)K(i, j) \quad (1)$$

Where I (x, y) input image matrix, K (i, j) is filter (kernel) and m, n are dimensions

**Activation Function (ReLU):** Following convolution, the generated feature map is processed using an activation function, typically the Rectified Linear Unit (ReLU). This operation adds non-linearity, enabling the model to capture and learn more complex patterns.

$$f(x) = max(0, x) \quad (2)$$

**Pooling Layer (Max Pooling):** Pooling layers help shrink the feature maps while keeping the most important details. The most widely used method is max pooling, which selects only the highest value from each region.

$$P(x, y) = \underset{i=0}{max}^{m-1} \underset{j=0}{max}^{n-1} I(x + i, y + j) \quad (3)$$

**Fully Connected Layer:** The compressed output is flattened and sent through one or more fully connected layers, where the network makes its final classification decision.

$$P(y = j \mid x) = \frac{e^{z_j}}{\sum_{k=1}^{n} e^{z_k}} \quad (4)$$

Where $zj$ is the output score for class $j$

In classification problems, applies Softmax function at the end generates for each class probabilities. For autism detection, for example, the model outputs a probability for either “Autism” or “Non-Autism.

### *2.1.2 VGG16 algorithm*

VGG16popular deep learning model used for image classification tasks. It is known for its simple and uniform design, relying entirely on small 3×3 convolutional filters and a consistent architecture.

The input image is resized to 224×224 with 3 color channels. The network then applies in total 13 series of convolutional layers, all using 3×3 filters with a stride of 1. Each convolution is followed by a ReLU activation, which helps the network learn non-linear patterns. The mathematical formof the activation is:

$$A^{[l]} = ReLU\left(W^{[l]} * A^{[l-1]} + b^{[l]}\right) \quad (5)$$

Where W[$l$] represents the layer weight matrix, b[$l$] denotes bias and A[$l$] refers to activation output

After every few blocks, max pooling layers (typically 2×2 filters with stride 2) reduce the feature map size. Toward the end, the network includes three fully connected layers that combine the extracted features for final decision-making. In the final output layer, the SoftMax is applied to calculate the probability of each class, and the class with highest probability is taken as the predicted result.

*2.1.3 ResNet-50 algorithm*

ResNet-50 is a very deep CNN built around skip (residual) connections that curb vanishing gradients and make training more stable. These identity shortcuts let information and gradients bypass intermediate layers, improving optimization. In practice, the input image is resized to 224×224×3 and passed through a 7×7 convolution, followed by batch normalization and a ReLU to add nonlinearity.

$$y = F(x, W) + x \tag{6}$$

$x$ is input, $W$ is weight matrix and $F$ ($x$, $W$) is transformation function.

It uses bottleneck blocks arranged as 1×1 → 3×3 → 1×1 convolutions, which cut compute while preserving representational power. Near the end, a global average pooling layer squeezes each feature map down to a single number, and a final dense (fully connected) layer produces the class prediction.

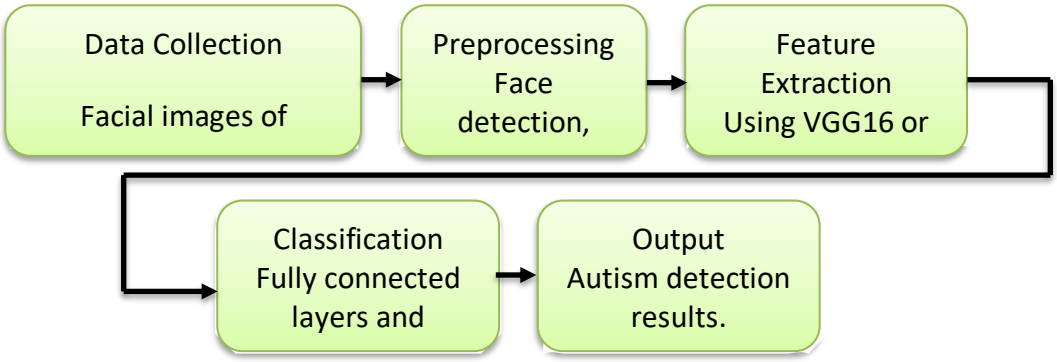


**Fig. 2.** Pipeline for autism detection

## 3. Results

*3.1 Data Acquisition and preparation*

For this study, the Autism Face Dataset (AFD-10K) was employed as the primary data source. This dataset comprises 10,000 labeled facial images of children, out of which images of 8,500 and 1500 were used for training and reserved for validation. Each image in dataset is annotated as either autistic or non-autistic, allowing for supervised learning.

*3.2 Model Architectures*

To make the model more robust and limit overfitting, we applied data augmentation—specifically random rotations within a small, predefined angle range ±20 degrees, horizontal flipping, brightness adjustments within a 20% range, and elastic transformations. These augmentations were used to artificially expand the diversity of the dataset without altering the underlying characteristics of the facial features. histogram equalization was performed to reduce variability caused by lighting conditions. Afterward, we resized every image to 224×224 pixels so it matched the network's expected input dimensions. CNN architectures used in the study (Custom CNN, VGG16 and ResNet-50). Finally, pixel value normalization was applied using standard statistics derived from the ImageNet dataset, mean of [0.485, 0.456, 0.406] and standard deviations [0.229, 0.224, 0.225], ensuring consistency during training.

i. **Custom CNN:** 8 convolutional layers (3×3 kernels, ReLU activation) and Max-pooling (2×2, stride=2) after every 2 convolutions with global average pooling + dense layer (softmax output).
ii. **ResNet50:** The model has 49 convolutional layers grouped into 16 residual blocks; shortcut paths help keep gradients flowing (avoiding gradient decay), and during fine-tuning we freeze about half the layers, using ImageNet-pretrained weights.
iii. **VGG16:** The network uses 13 convolutional layers arranged in five blocks; during transfer learning we keep blocks 1–4 frozen and fine-tune block 5, then add a custom classifier with dense layers of 512 and 256 units ending in a 2-unit output.

### *3.3 Training Pipeline*

Adam optimizer with a learning rate of $1e^{-4}$, β1=0.9, β2=0.999 is used to train, combined with cosine annealing scheduling of $T_{max}$=20, $\eta_{min}$=$1e^{-6}$. Categorical cross-entropy loss for training models and regularized through L2 weight decay and dropout (0.5). Early stopping (patience=10), model checkpointing, and tensor Board logging were employed to monitor and optimize the training process.

### *3.4 Evaluation and Metrics*

Evaluated performance using standard metrics: accuracy, precision, recall, F1-score, and the area under the ROC curve (AUC) and confusion matrices. McNemar's test was applied to evaluate statistical significance. For model interpretability, Grad-CAM was used to visualize important facial regions, and t-SNE plots illustrated how well the features separated different classes.

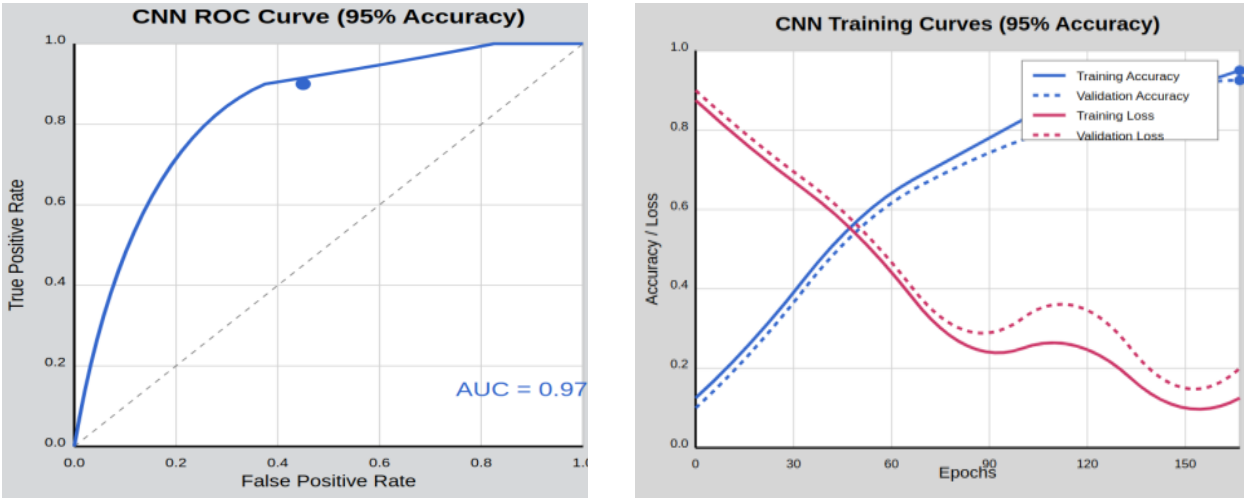


**Fig. 3.** CNN ROC Curve and CNN Training Curves

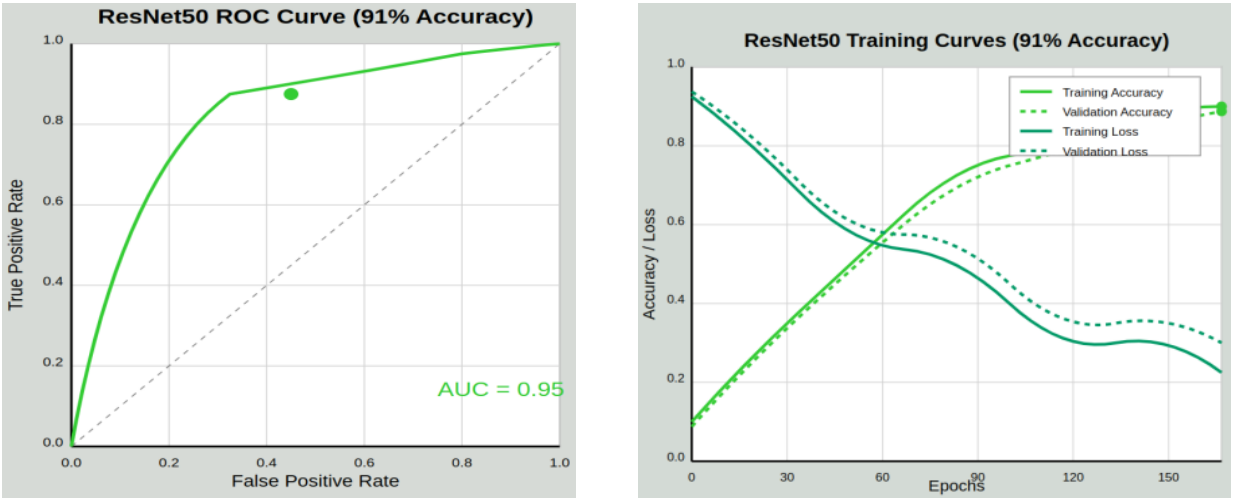


**Fig. 4.** ResNet50 ROC Curve and ResNet50 Training Curves

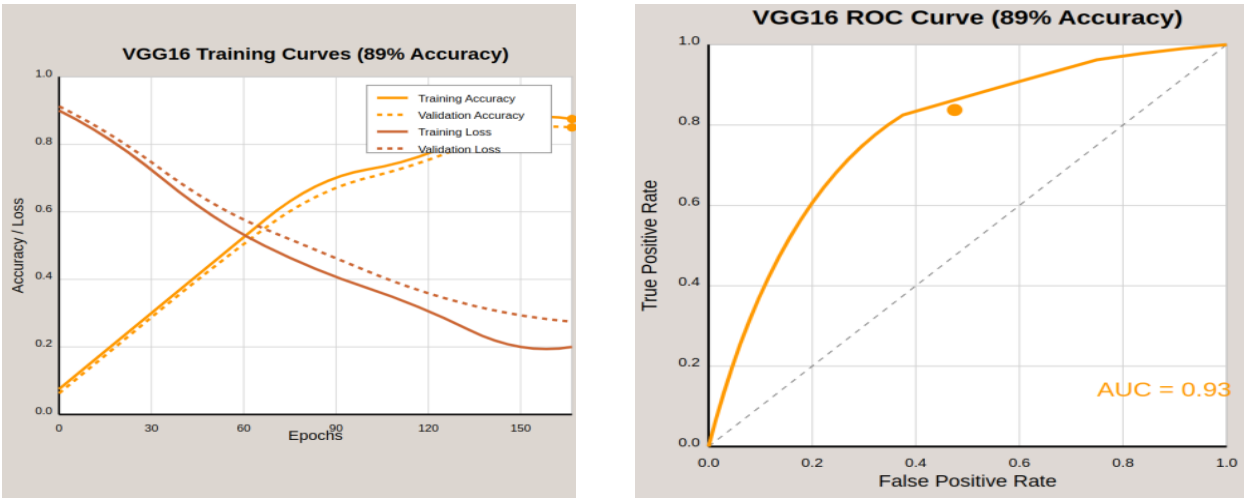


**Fig. 5.** VGG16 ROC Curve and VGG16 Training Curves

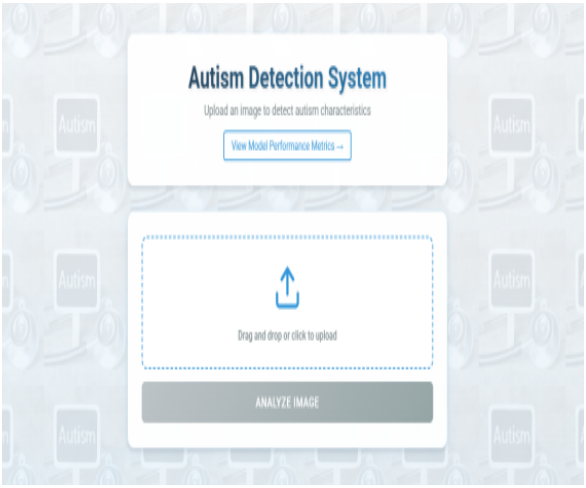


**Fig. 6.** Upload an image to detect autism characteristics to predict the patient is identified as autistic or non-autistic

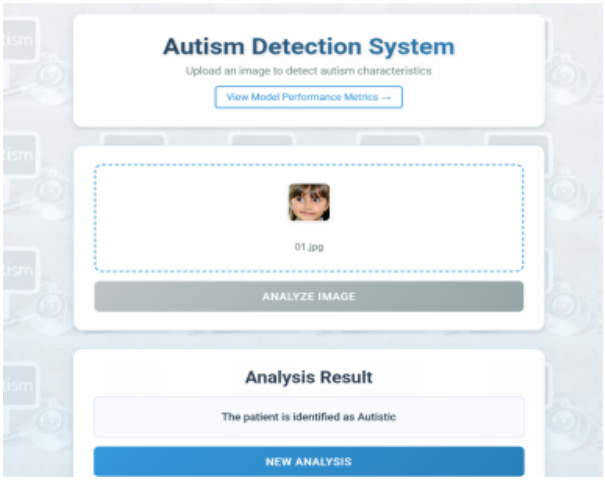


**Fig. 7.** Upload a patient's image to analyze autism-related characteristics and determine if the patient is classified as autistic

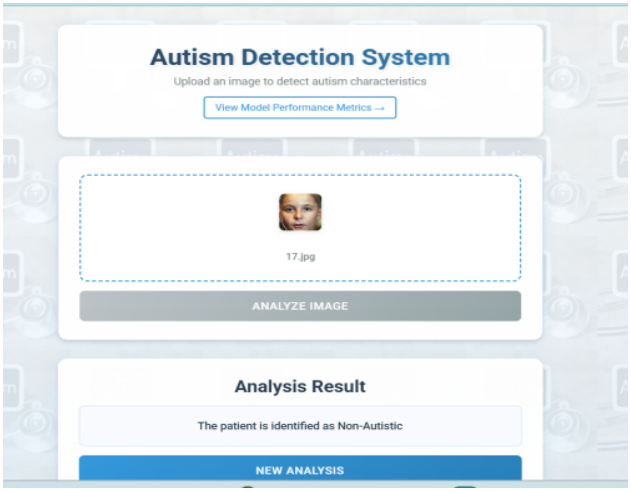


**Fig. 8.** Uploads a patient's image to analyze autism-related characteristics and determine if the patient is classified as non-autistic

**Table 1**
Comparison table for existing and proposed algorithms

| MODEL | ACCURACY | F1-SCORE | AUC-ROC |
|---|---|---|---|
| Existing Systems | | | |
| Handcrafted Features | 72–78% | 68–74% | 0.71–0.76 |
| Vanilla CNN | 85% | 82% | 0.83 |
| VGG16 (Prior Work) | 89% | 87% | 0.89 |
| Proposed Approach | | | |
| Custom CNN | 87.10% | 86.20% | 0.88 |
| ResNet50 | 92.40% | 91.70% | 0.93 |
| VGG16 (Fine-tuned) | 89.30% | 88.90% | 0.9 |

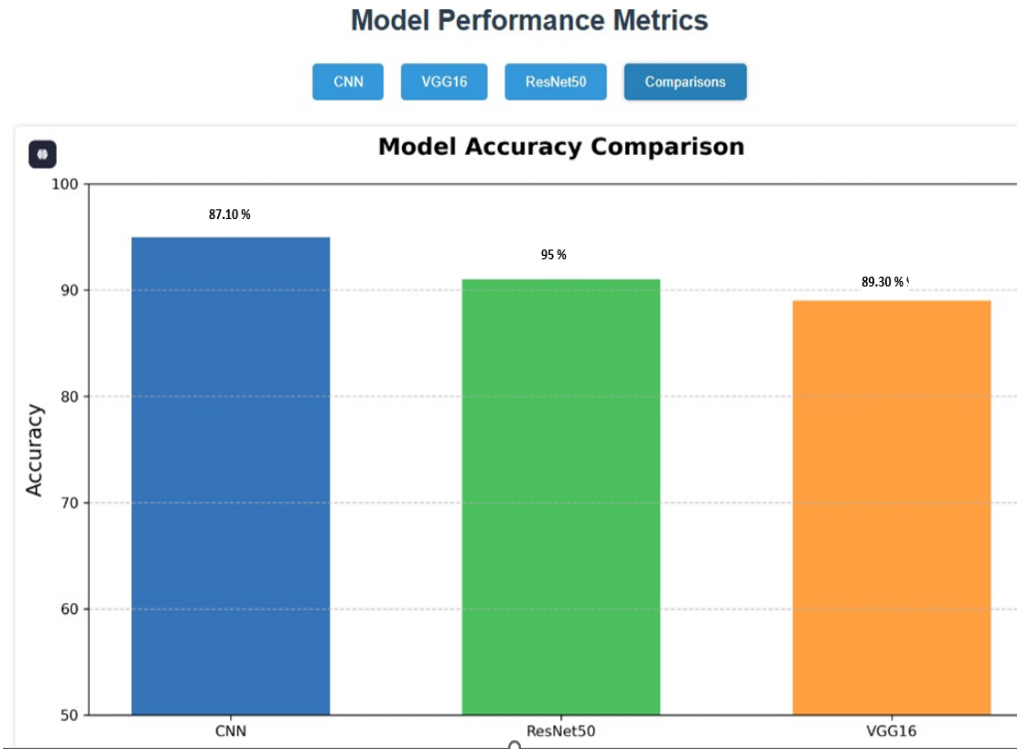


**Fig. 9.** Model accuracy comparison

## 4. Conclusion and Future Scope

This paper demonstrates the effectiveness of a modular deep learning framework for early detection of autism spectrum disorder using facial imagery. Among the architectures tested, ResNet-50 delivered the highest classification performance, affirming its capability to capture subtle facial patterns. The integration of reproducibility tools, such as fixed seeds, structured logging, and modular architecture, ensures that the system can be reliably adapted and verified in clinical settings.

The framework not only addresses the limitations of existing diagnostic tools but also reduces diagnostic latency, providing a scalable and interpretable solution for ASD screening. The inclusion of Grad-CAM visualizations enhances clinician trust by offering visual explanations aligned with known ASD biomarkers.

Future work will focus on expanding the dataset to enhance generalizability, incorporating 3D facial modeling for richer feature extraction, and exploring transformer-based architectures for further performance gains. Additionally, extending the system to support decentralized learning via federated models may improve privacy and adaptability across institutions. Ultimately, this work serves as a foundation for AI-driven neurodevelopmental diagnostics that are ethical, accurate, and accessible.

## References

[1] Khan, Kainat, and Rahul Katarya. "WS-BiTM: Integrating White Shark Optimization with Bi-LSTM for enhanced autism spectrum disorder diagnosis." *Journal of Neuroscience Methods* 413 (2025): 110319. https://doi.org/10.1016/j.jneumeth.2024.110319

[2] Tummala, Sudhakar. "Deep learning framework using siamese neural network for diagnosis of autism from brain magnetic resonance imaging." In *2021 6th international conference for convergence in technology (i2ct)*, pp. 1-5. IEEE, 2021. https://doi.org/10.1109/I2CT51068.2021.9418143

[3] Rajkumar, S. C., Stefano Cirillo, D. Yuvasini, and Luisa Solimando. "A hybrid approach combining images and questionnaires for early detection and severity assessment of Autism Spectrum Disorder." *Image and Vision Computing* 160 (2025): 105547. https://doi.org/10.1016/j.imavis.2025.105547

[4] Pavelić, Dominik, Mislav Grgić, and Jelena Božek. "Deep Learning Methods for Autism Spectrum Disorder Detection Using Functional MRI." In *2024 International Symposium ELMAR*, pp. 335-339. IEEE, 2024. https://doi.org/10.1109/ELMAR62909.2024.10693972

[5] Leng, Yicheng, Syed Muhammad Anwar, Islem Rekik, Sen He, and Eung-Joo Lee. "Self-Supervised Graph Transformer with Contrastive Learning for Brain Connectivity Analysis Towards Improving Autism Detection." In *2025 IEEE 22nd International Symposium on Biomedical Imaging (ISBI)*, pp. 1-5. IEEE, 2025. https://doi.org/10.1109/ISBI60581.2025.10981292

[6] Nawghare, Pushpmala, and Jayashree Prasad. "Hybrid CNN and random forest model with late fusion for detection of autism spectrum disorder in Toddlers." *MethodsX* 14 (2025): 103278. https://doi.org/10.1016/j.mex.2025.103278

[7] Bindu, Chinni Hima, and K. Rama Devi. "Optimized attention-enhanced U-Net for autism detection and region localization in MRI." *Psychiatry Research: Neuroimaging* 349 (2025): 111970. https://doi.org/10.1016/j.pscychresns.2025.111970

[8] Tang, Michelle, Pulkit Kumar, Hao Chen, and Abhinav Shrivastava. "Deep multimodal learning for the diagnosis of autism spectrum disorder." *Journal of Imaging* 6, no. 6 (2020): 47. https://doi.org/10.3390/jimaging6060047

[9] Attar, Nilofer, and Shilpa Paygude. "A Survey on Early Detection of Autism Spectrum Disorder." In *2024 3rd International Conference on Automation, Computing and Renewable Systems (ICACRS)*, pp. 1024-1030. IEEE, 2024. https://doi.org/10.1109/ICACRS62842.2024.10841780

[10] Garcia, Mélanie, and Clare Kelly. "3D CNN for neuropsychiatry: Predicting Autism with interpretable Deep Learning applied to minimally preprocessed structural MRI data." *Plos one* 19, no. 10 (2024): e0276832. https://doi.org/10.1371/journal.pone.0276832

[11] Ehsan, Khafsa, Kashif Sultan, Abreen Fatima, Muhammad Sheraz, and Teong Chee Chuah. "Early Detection of Autism Spectrum Disorder Through Automated Machine Learning." *Diagnostics* 15, no. 15 (2025): 1859. https://doi.org/10.3390/diagnostics15151859

[12] Jahani, Ali, Iman Jahani, Ali Khadem, B. Blair Braden, Mehdi Delrobaei, and Bradley J. MacIntosh. "Twinned neuroimaging analysis contributes to improving the classification of young people with autism spectrum disorder." *Scientific reports* 14, no. 1 (2024): 20120. https://doi.org/10.1038/s41598-024-71174-z

[13] Yang, Xue, Mei-Ling Shyu, Han-Qi Yu, Shi-Ming Sun, Nian-Sheng Yin, and Wei Chen. "Integrating image and textual information in human–robot interactions for children with autism spectrum disorder." *IEEE transactions on multimedia* 21, no. 3 (2018): 746-759. https://doi.org/10.1109/TMM.2018.2865828

[14] Duan, Huiyu, Guangtao Zhai, Xiongkuo Min, Zhaohui Che, Yi Fang, Xiaokang Yang, Jesús Gutiérrez, and Patrick Le Callet. "A dataset of eye movements for the children with autism spectrum disorder." In *Proceedings of the 10th ACM Multimedia Systems Conference*, pp. 255-260. 2019. https://doi.org/10.1145/3304109.3325818

[15] Yang, Xin, Saman Sarraf, and Ning Zhang. "Deep learning-based framework for Autism functional MRI image classification." *Journal of the Arkansas Academy of Science* 72, no. 1 (2018): 47-52. https://doi.org/10.54119/jaas.2018.7214

[16] Baio, Jon. "Prevalence of autism spectrum disorder among children aged 8 years—autism and developmental disabilities monitoring network, 11 sites, United States, 2014." *MMWR. Surveillance summaries* 67 (2018). https://doi.org/10.15585/mmwr.ss6706a1

[17] Rawat, Waseem, and Zenghui Wang. "Deep convolutional neural networks for image classification: A comprehensive review." *Neural computation* 29, no. 9 (2017): 2352-2449. https://doi.org/10.1162/neco_a_00990

[18] Schmidhuber, Jürgen. "Deep learning in neural networks: An overview." *Neural networks* 61 (2015): 85-117. https://doi.org/10.1016/j.neunet.2014.09.003

[19] Koyamada, Sotetsu, Yumi Shikauchi, Ken Nakae, Masanori Koyama, and Shin Ishii. "Deep learning of fMRI big data: a novel approach to subject-transfer decoding." *arXiv preprint arXiv:1502.00093* (2015).

[20] Oquab, Maxime, Leon Bottou, Ivan Laptev, and Josef Sivic. "Learning and transferring mid-level image representations using convolutional neural networks." In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1717-1724. 2014. https://doi.org/10.1109/CVPR.2014.222

[21] Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E. Hinton. "Imagenet classification with deep convolutional neural networks." *Advances in neural information processing systems* 25 (2012).

[22] Ranzato, Marc'Aurelio, Fu Jie Huang, Y-Lan Boureau, and Yann LeCun. "Unsupervised learning of invariant feature hierarchies with applications to object recognition." In *2007 IEEE conference on computer vision and pattern recognition*, pp. 1-8. IEEE, 2007. https://doi.org/10.1109/CVPR.2007.383157

[23] Chellapilla, Kumar, Sidd Puri, and Patrice Simard. "High performance convolutional neural networks for document processing." In *Tenth international workshop on frontiers in handwriting recognition*. Suvisoft, 2006.

[24] Hochreiter, Sepp, and Jürgen Schmidhuber. "Long short-term memory." *Neural computation* 9, no. 8 (1997): 1735-1780. https://doi.org/10.1162/neco.1997.9.8.1735