Autism Spectrum Disorder Detection in Children Using Convolutional Neural Networks and Facial Image Analysis

Nidhin Raj J S (25AI60R01)
Mtech Artificial Intelligence
IIT Kharagpur
Email: xxnidhin25@kgpian.iitkgp.ac.in

Ravindra Mina (25AI60R02)
Mtech Artificial Intelligence
IIT Kharagpur
Email: ravindramina25@kgpian.iitkgp.ac.in

Abstract—**Problem Statement: This project addresses the problem of early detection of Autism Spectrum Disorder (ASD) using facial image analysis. Clinical diagnosis typically relies on behavioral assessments and questionnaires, which are subjective and time consuming. Subtle visual patterns in facial features and expressions can provide objective markers for ASD, making computer vision techniques suitable for automated analysis.**

Motivation: Early identification of ASD is critical for timely intervention and improved developmental outcomes. A vision-based system enables scalable, non-invasive, and cost-effective screening, particularly in settings with limited access to specialists. By analyzing facial images automatically, we aim to create a robust visual computing tool that supports healthcare professionals.

Challenges: The primary challenges include variability in facial appearance across age, gender, and ethnicity, limited dataset size, and subtle differences between classes that make feature extraction complex. Developing a model capable of learning discriminative visual patterns while generalizing to unseen faces is a key challenge.

Data Requirement: The project uses a facial image dataset consisting of individuals with and without ASD. We utilize an available Kaggle (or zenodo) facial image dataset curated for autism spectrum research. Preprocessing steps include face detection, alignment, normalization, and augmentation to enhance feature consistency and improve model performance.

Techniques/Algorithms: The methodology combines multiple stages of visual computing. First, image preprocessing techniques such as face detection, alignment, and normalization are applied to ensure consistency. Next, visual feature extraction is performed using Convolutional Neural Networks (CNNs). Finally, these extracted visual features are used for binary classification of ASD versus non-ASD cases. This end-to-end pipeline highlights the role of computer vision in processing and analyzing facial imagery for healthcare applications.

Evaluation: Performance will be measured using accuracy, F1-score, and confusion matrix analysis.

Impact: This project demonstrates the application of visual computing techniques to healthcare diagnostics. By automating ASD screening from facial images, it aims to provide a fast, scalable, and objective support tool, showcasing the potential of computer vision in medical image analysis.

Index Terms—**Autism Spectrum Disorder, CNN, SmoothGrad-CAM++, interpretability, face images, deep learning**

I. Introduction

Autism Spectrum Disorder (ASD) affects social communication and behaviour and often appears early in childhood. Accurate early screening can greatly improve outcomes by enabling early therapy. Traditional diagnostic processes are clinician-led and time-consuming. Automated screening using computer vision provides a low-cost, scalable adjunct to clinical assessment and may flag children who need further evaluation.

In this project we focus on face-image based screening using a Kaggle face dataset of children (autistic / non-autistic). Our objective: build a reproducible pipeline — train a CNN, evaluate it, and provide per-sample visual explanations using SmoothGrad-CAM++ so clinicians / researchers can inspect model reasoning.

II. Related Work

Automated ASD detection research can be grouped into (i) questionnaire / behavioral-feature approaches and (ii) image-based deep learning. Representative studies:

A. Questionnaire/Tabular Methods

Suman Raj et al. (2020) compared Naïve Bayes, SVM, Logistic Regression, KNN, ANN and CNN on UCI screening datasets (Children: 292 samples; Adolescents: 104; Adults: 704). They reported high accuracy metrics for CNNs on those tabular sets (reported up to 99% for adults). These experiments emphasize ML utility on structured screening data but do not use images or clinical cohorts, limiting generalization.

B. Face-image Deep Learning

Rashid & Shaker (2023) applied transfer learning (Xception, VGG16) to the Kaggle face dataset and reported Xception produced best results (91%). Arumugam et al. (2021) trained a custom CNN (from scratch) on the same dataset reporting similar accuracy. Both image-based studies comment on dataset noise, label quality and the need for clinical validation.

C. Synthesis

Tabular screening and image-based screening are complementary—questionnaire data capture behaviour while images may encode subtle facial cues. Across works, the main limitations are small/non-clinical datasets, potential dataset bias, and limited external validation. Our work builds on these by adding interpretability (SmoothGrad-CAM++) and reporting full metrics and visual examples.

III. Dataset and Preprocessing

A. Dataset

We used the Kaggle *Autistic Children Face Dataset* (public). The dataset contains approximately **2,940 images** spanning both classes. Following common practice we split into train/validation/test subsets (we used an 70/15/15 split for experiments). This yields roughly:

- Train: ≈2536 images
- Validation: ≈100 images
- Test: ≈300 images

B. Preprocessing

All images are resized to 224×224. For training we used on-the-fly augmentation: horizontal flips, random rotations (±20°) and normalization to the [0,1] range for the custom CNN. Validation and test images receive only resizing and normalization.

IV. Methodology

A. Model architecture

We implemented a compact CNN (three convolutional blocks followed by a 128-unit dense layer and dropout). The architecture mirrors the one used in earlier TensorFlow experiments to enable weight parity; details:

- Conv1: 32 filters, kernel 3×3, padding=1, ReLU + MaxPool
- Conv2: 64 filters, kernel 3×3, padding=1, ReLU + MaxPool
- Conv3: 128 filters, kernel 3×3, padding=1, ReLU + MaxPool
- FC1: 128 units, ReLU, Dropout(0.4)
- Output: 2 logits (CrossEntropyLoss)

B. Training regimen

- Optimizer: Adam, lr=1e-4
- Loss: CrossEntropy
- Batch size: 32
- Epochs: 10

We track training and validation loss and accuracy each epoch and save final weights to a checkpoint file (e.g., `autism_cnn.pt`) for reproducible inference.

C. Interpretability: SmoothGrad-CAM++

To make model decisions inspectable we use **SmoothGrad-CAM++** (torchcam). For each image we compute the CAM++ map for the predicted class and visualise:

1) the original image,
2) the raw CAM heatmap,
3) an overlay of heatmap on image.

We ensure the final conv block's parameters have `requires_grad=True` and the input tensor used for CAM hooks has `requires_grad=True` as required by torchcam.

V. Experimental Results

A. Training and Validation Curves

Figure 1 shows the training and validation losses and accuracies across epochs. These plots help assess convergence and potential overfitting.

B. Test Performance

After training, we evaluated on the held-out test set and obtained:

TABLE I: Test set metrics (classification report)

Class	Precision	Recall	F1-score	Support
autistic	0.75	0.84	0.79	150
non_autistic	0.82	0.73	0.77	150
accuracy	0.78333 (300 samples)			

(These values are the test numbers from our run: Test Accuracy = 0.783333.)

C. Confusion Matrix

Figure 2 displays the confusion matrix on the test set. The model makes balanced errors across classes; inspection of misclassified examples via CAM can reveal causes such as occlusions or atypical poses.

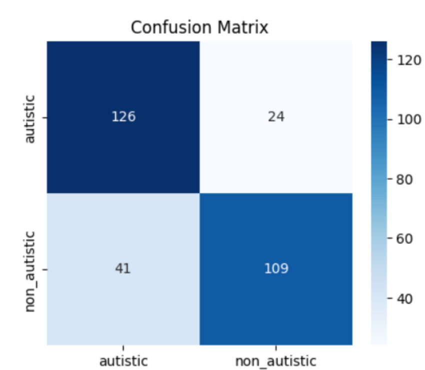
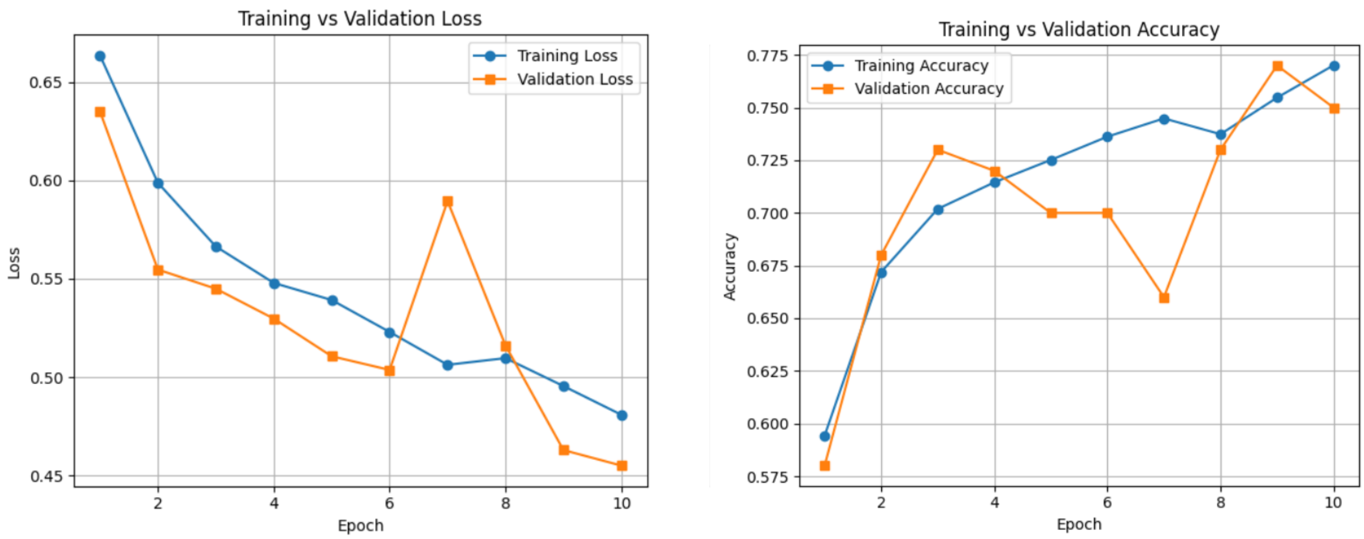


Fig. 2: Confusion matrix on the held-out test set



(a) Training vs Validation Loss

(b) Training vs Validation Accuracy

Fig. 1: Learning curves.

D. Qualitative Interpretability

Representative visualisations for an *autistic* and a *non-autistic* sample are shown in Figures 3 and 4. Observe that CAM highlights the peri-orbital and mid-face regions in many examples — consistent with models attending to gaze/eye-related cues.

VI. Discussion

The quantitative results (78.33% test accuracy) show the model learns discriminative cues but is not ready for clinical deployment. Key observations:

- **Generalization:** Performance on the public/test set is moderate; further gains likely need more data and stronger regularization or transfer learning.
- **Interpretability:** CAM maps show plausible attention regions (eyes / upper face), which provides partial validation that the model is not exploiting irrelevant artefacts.
- **Bias & limitations:** Dataset is crowdsourced and may contain demographic skews, label noise, or repeated subjects. These factors can inflate apparent performance or degrade trustworthiness.

VII. Conclusion and Future Work

We implemented a compact CNN for ASD screening from facial images and paired it with SmoothGrad-CAM++ interpretability. Results indicate the approach has promise as a low-cost screening aid but requires:

- training and testing on clinically validated, demographically diverse datasets,
- bias/fairness analysis across age/sex/ethnicity,
- multi-modal fusion (combine face images with behavioral/questionnaire features) to improve robustness,

Acknowledgements

We thank the dataset contributors on Kaggle and the open-source communities behind PyTorch and torchcam.

References

[1] S. Raj and S. Masood, “Analysis and Detection of Autism Spectrum Disorder Using Machine Learning Techniques”, *Procedia Computer Science*, 2020.

[2] A. F. Rashid and S. Shaker, “Autism Spectrum Disorder Detection Using Face Features Based on Deep Neural Network”, 2023.

[3] S. R. Arumugam et al., “Prediction of Autism Spectrum Disorder in Children Using Face Recognition”, ICOSEC 2021.

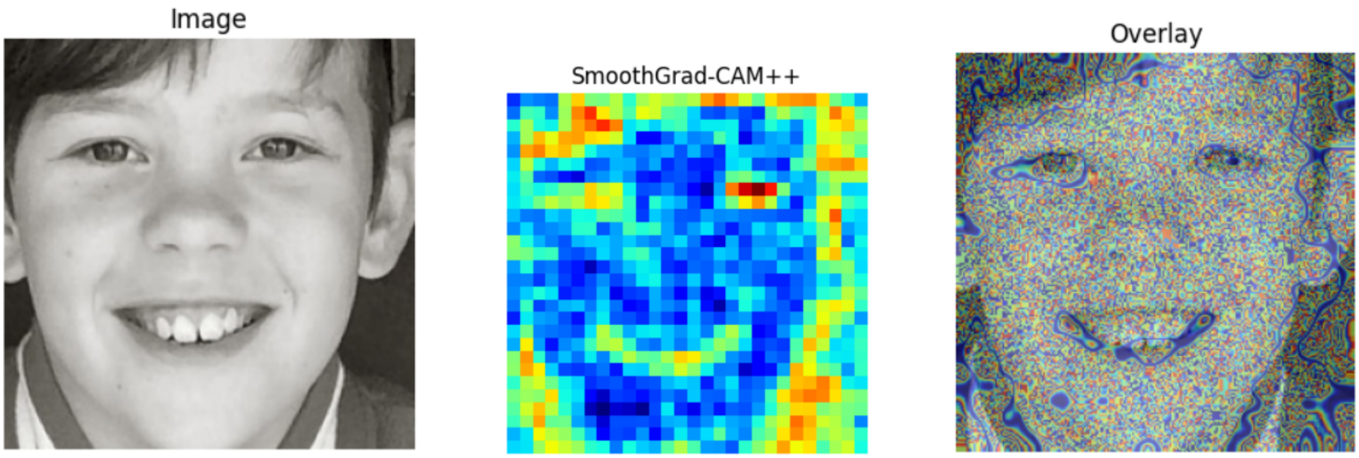
Appendix

Nidhin Raj J S (25AI60R01)

- Conceptualization, Literature review, Methodology, CNN Implementation, Training, Evaluation, Draft writing.

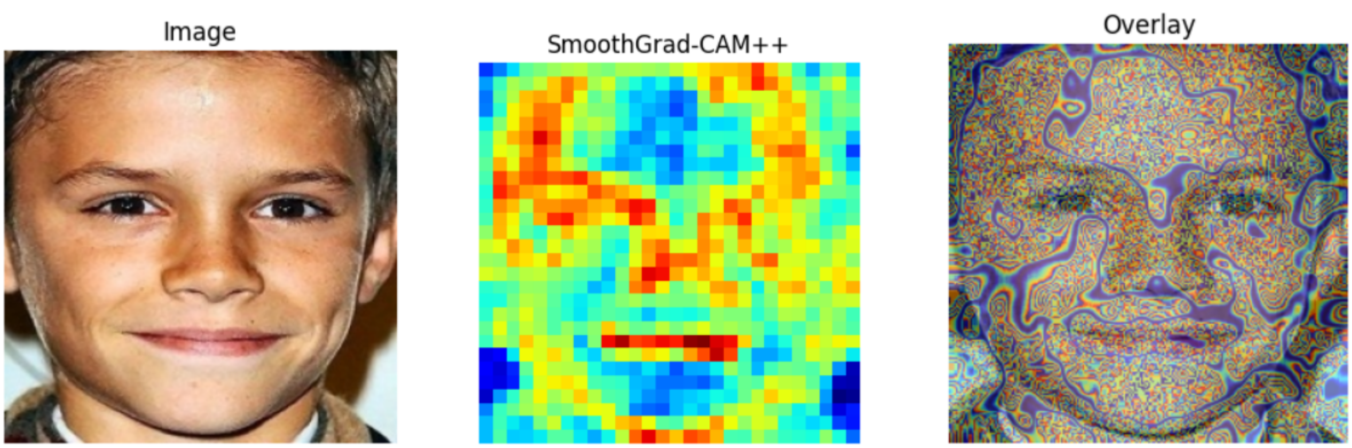
Ravindra Mina (25AI60R02)

- Smooth Grad cam++ Implementation, Final editing.



(a) Original (b) Smooth Grad CAM++ (c) Overlay

Fig. 3: Autistic sample: original, Smooth Grad CAM++ heatmap, and overlay.



(a) Original (b) Smooth Grad CAM++ (c) Overlay

Fig. 4: Non-autistic sample: original, Smooth Grad CAM++ heatmap, and overlay.