Jurnal Infotel
Vol. 17, No. 3, August 2025, pp. 612–626. doi:10.20895/infotel.v17i3.1363

Research Article

Mobile Application Development for Facial Classification of Autistic Children Based on MobileNet-V3

Irsyan Ramadhan[1], Melinda Melinda[2,*], Yunidar Yunidar[3], Donata D. Acula[4], Zulfan Zainal[5], Riska Miftahujjannah[6], and Siti Rusdiana[7]

[1,2,3,5,6]Department of Electrical Engineering and Computer, Universitas Syiah Kuala, Banda Aceh 23111, Indonesia
[4]Doctoral Degree in Mathematics Education, Department of Computer Science, College of Information and Computing Sciences, University of Santo Tomas, Manila 1008, Philippines
[7]Department of Mathematics, Universitas Syiah Kuala. Banda Aceh 23111, Indonesia

*Corresponding email: melinda@usk.ac.id

Received: May 15, 2025; Revised: July 04, 2025; Accepted: July 06, 2025.

Abstract: Early detection of autism spectrum disorder (ASD) plays a crucial role in interventions that can significantly improve children's cognitive and social development. Traditional diagnostic methods often rely on clinical observation, which may introduce subjectivity and delay diagnosis. This study presents a Flutter-based mobile application that integrates an on-device deep learning model for classifying facial images of autism. The classification model is built on the lightweight MobileNetV3-Small architecture, optimized for real-time inference on mobile devices using TensorFlow Lite. A dataset of 600 facial images (300 autistic, 300 non-autistic) was collected from local schools in Banda Aceh, Indonesia, and expanded to 1,860 images using augmentation techniques. The model was trained using transfer learning and achieved 97% accuracy, precision, recall, and F1 Score on unseen test data. Users can upload images through camera or gallery, with classification results displayed in 1 to 2 seconds and stored in Firebase for history tracking. The proposed system provides a fast, accessible, and cost-effective tool for the detection of ASD. Unlike many prior studies, this work emphasizes full on-device execution without server dependency, improving both speed and privacy. However, since the dataset was collected from a single location with limited demographic diversity, more validation is needed on broader, multisite datasets to ensure generalizability and fairness in future deployment.

Keywords: Autism Spectrum Disorder, Facial Image Classification, MobileNetV3, Flutter, Mobile Application, Firebase

1 Introduction

Autism Spectrum Disorder (ASD) is a complex neurodevelopmental condition that affects communication, behavior, and social interaction. According to the World Health Organization (WHO), approximately 1 in 160 children worldwide is diagnosed with ASD, although the actual figure may be higher in underserved regions due to limited access to diagnostic services [1]. Early detection plays a vital role in enabling timely interventions, such as behavioral and communication therapies, which have been shown to significantly improve developmental outcomes and quality of life for affected children [2]. Studies have consistently shown that children who receive early intervention exhibit better cognitive and behavioral performance compared to those diagnosed later [3,4].

Despite its importance, the diagnosis of ASD remains a challenge in many parts of the world. Standard diagnostic tools such as the Autism Diagnostic Observation Schedule (ADOS) and the Autism Diagnostic Interview-Revised (ADI-R) rely heavily on expert observation and parental reporting, which introduces potential subjectivity and delays in identification [5]. Furthermore, currently there is no established biological marker for ASD, making the diagnosis highly dependent on qualitative evaluations. This gap underscores the urgent need for more objective, data-driven, and scalable screening tools.

In recent years, deep learning has emerged as a powerful tool in medical image analysis, including ASD detection. Convolutional neural networks (CNN) are particularly suited for this task due to their ability to extract complex features from images, including subtle facial characteristics that can distinguish children with ASD from their neurotypical peers [6]. Previous research has shown that CNNs can achieve high accuracy in classifying facial expressions and detecting neurodevelopmental conditions. Among the various CNN architectures, MobileNet stands out due to its efficiency and low computational cost, which makes it suitable for mobile deployment without requiring cloud processing [7].

Although several studies have applied CNNs for ASD detection using facial images, most existing systems rely on server-based processing or high-end hardware. This approach may not be practical in low-resource settings. To address this limitation, we propose a lightweight, fully on-device solution that leverages the MobileNetV3-Small architecture integrated within a mobile application built using Flutter. This design enables fast inference and preserves user privacy by eliminating the need for external servers. Users can upload a child's facial image via the camera or gallery, and the mobile application returns a classification result within seconds.

The dataset used in this study comprises 600 original facial images of children (300 with ASD and 300 without), collected from special education schools and elementary schools in Banda Aceh, Indonesia. To improve data diversity and reduce overfitting, augmentation techniques such as horizontal flipping, Gaussian noise, and contrast adjustment were applied, expanding the dataset to 1,860 images. The model was trained using transfer learning, employing ImageNet pre-trained weights, and fine-tuned for binary classification. Training used a 70/15/15 split for training, validation and testing, respectively, and achieved 97% accuracy, precision, recall, and F1-score on unseen test data. This research contributes three key innovations:

- A lightweight and accurate facial image classifier for ASD detection using MobileNetV3-Small;
- Integration of the classifier into a cross-platform mobile application that supports real-time, offline inference using TensorFlow Lite;

- A practical tool to support ASD screening in underserved communities where specialist diagnostic services may be unavailable.

Although the results are promising, the data and the literature are limited in terms of geographic and demographic diversity, potentially affecting generalizability. Future work should focus on multisite data collection, validation with benchmark datasets, and real-world usability testing in collaboration with clinicians.

2 Research Method

This study aims to develop a Flutter-based mobile application for classifying facial images of autistic and non-autistic children using the MobileNetV3-Small architecture. The research workflow includes five main stages: (1) data collection, (2) preprocessing, (3) model training and validation, (4) mobile application integration, and (5) system evaluation. Firebase Firestore is used for user authentication and storing classification history. The overall workflow is summarized in Figure 1, which illustrates the step-by-step process from data acquisition to mobile deployment.

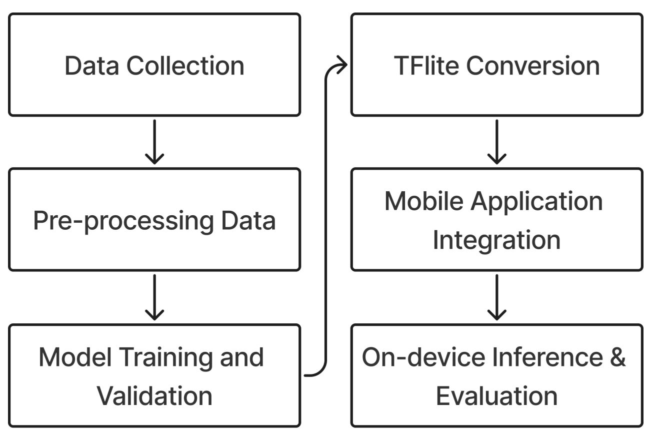

Figure 1: Workflow of the proposed system.

2.1 Data Collection

Facial image data were collected from two educational institutions in Banda Aceh, Indonesia: a special needs school and MIN 8 Banda Aceh. Ethical clearance was obtained (Approval No. 036/EA/FK/2025), and data collection was carried out with teacher assistance to ensure student comfort. Informed consent was obtained from all parents or legal guardians.

A total of 600 facial images were collected from children aged 6–14 years, with an equal split of 300 autistic and 300 non-autistic samples, where the example dataset is presented in Figure 2. The photos were taken in a controlled indoor environment (4×3 m classroom, 30W lighting, 25˘27°C, 40–45% humidity) during morning hours (9:00–12:00) when students were cooperative. Controlled acquisition helps reduce environmental noise in deep learning models [8].

(a) (b)

Figure 2: Sample Image: (a) Autistic, (b) non-Autistic

2.2 Preprocessing Data

Preprocessing is a crucial step in improving image quality and improving the accuracy of deep learning models [8]. The pipeline (see Figure 3) involved several stages:

1. Cropping: The face region was extracted using Python's PIL and NumPy libraries [9], removing irrelevant background details.
2. Resizing: All images were resized to 224×224 pixels to match the MobileNetV3 input standard [10].
3. Augmentation: To increase dataset diversity and minimize overfitting, augmentation techniques such as horizontal flipping, Gaussian noise, and contrast adjustment were applied [8], [11].

The final dataset increased to 1,860 images (see Table 1). A stratified split was applied using a 70:15:15 ratio for training, validation, and testing while maintaining class balance.

[!htb]

Table 1: Dataset distribution for training, validation, and testing

Datasets	**Autistic**	**Non-Autistic**	**Total**
Training	840	840	1680
Validation	45	45	90
Testing	45	45	90

2.3 Model Development

The classification model was built using the MobileNetV3-Small architecture, chosen for its optimal trade-off between performance and computational efficiency, particularly in

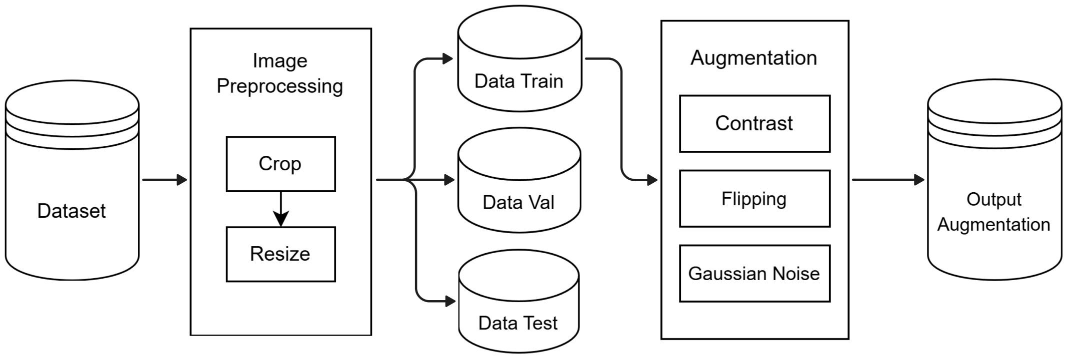

Figure 3: Preprocessing data flowchart.

mobile deployments [12]. It contains significantly fewer parameters than deeper networks such as ResNet or Inception, making it ideal for real-time applications on edge devices [13].

Transfer learning was employed by initializing the model with ImageNet weights and making all layers trainable to fine-tune performance on the ASD classification task [13]. The training process used:

- Binary Cross-Entropy loss function, suitable for binary classification problems;
- Stochastic Gradient Descent (SGD) optimizer, with a low learning rate of 0.001 for stability [14];
- Model checkpoints, which saved the best model weights based on validation loss.

Table 2: Model training hyperparameters.

Parameter	**Value**
Architecture	MobileNet-v3 Small
Loss Function	cross entropy
Optimizer	SGD
Epoch	200
Learning Rates	10^{-3}
Batch Size	16

The final model was selected based on the lowest validation loss and all relevant training metrics were documented for evaluation.

2.4 Model Evaluation

The classification model was evaluated using a confusion matrix, which provides a detailed view of the model's ability to distinguish between autistic and non-autistic classes. The confusion matrix was used to compute four key metrics commonly used in binary classification tasks: accuracy, precision, recall, and F1 Score, defined as follows [15]:

$$\text{Accuracy} = \frac{(\text{TP} + \text{TN})}{(\text{TP} + \text{FP} + \text{FN} + \text{TN})} \tag{1}$$

$$\text{Precision} = \frac{(\text{TP})}{(\text{TP} + \text{FP})} \tag{2}$$

$$\text{Recall} = \frac{(\text{TP})}{(\text{TP} + \text{FN})} \tag{3}$$

$$\text{F1 Score} = 2 \times \frac{(\text{Precision} \times \text{Recall})}{(\text{Precision} + \text{Recall})} \tag{4}$$

where TP is True Positive, TN is True Negative, FP is False Positive, and FN is False Negative. These metrics offer a comprehensive assessment of classification reliability, particularly for healthcare-related applications, where both false positives and false negatives can have significant implications.

2.5 TFLite Conversion and Integration

To enable real-time inference on mobile devices, the trained MobileNetV3-Small model was exported in *.h5 format and converted to TensorFlow Lite (TFLite). This format significantly reduces model size and computation requirements, making it suitable for local inference [14].

By integrating TFLite into the mobile application, classification can be performed entirely on-device, without requiring cloud access. This approach improves inference speed (1–2 seconds per image), reduces dependency on network stability, and ensures better privacy protection essential for use in educational and clinical settings.

2.6 Mobile Application Development

Figure 4 illustrates the mobile application development process, which involves several stages: Planning, Design, Development, and Testing.

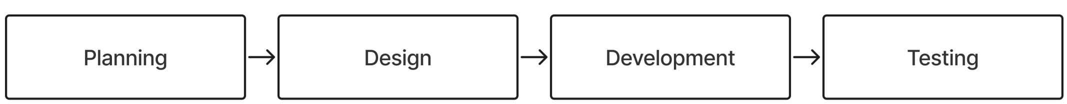

Figure 4: Flowchart mobile application development.

2.6.1 Planning

The workflow between the front-end, back-end, and cloud services is illustrated in Figure 5. The front-end was developed using Flutter to ensure a consistent and responsive user interface across devices [16]. The backend handles real-time classification using the locally embedded TFLite model, which supports fast inference and enhances user privacy by avoiding cloud-based processing [17]. Classification results are stored as diagnostic history in the Firebase Cloud Firestore, which also serves as the application back-end infrastructure for user data management [18]. A use case diagram was also created to describe interactions between users and the system, forming the basis for functional planning.

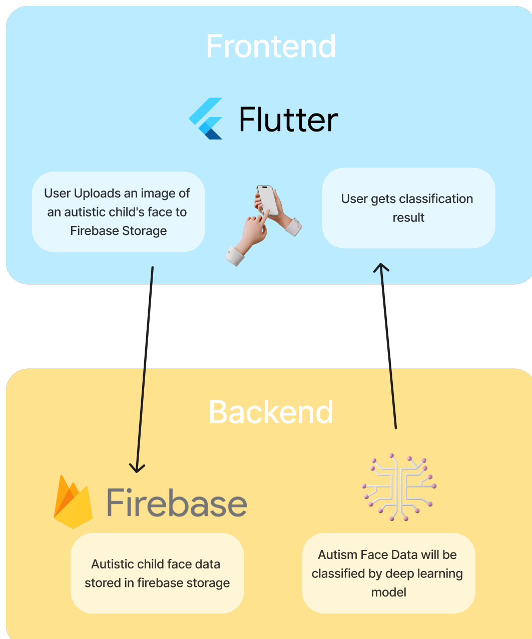

Figure 5: Schematic of the autism child classification mobile application.

A use case diagram was created to define user interactions. Users can register, log in, upload images, and receive classification results immediately. All diagnostic results are saved in the Firebase database for later reference.

2.6.2 Design

Figma tools create wireframes and prototypes in the design process to validate the application layout prior to development [19]. This design included the main screen, the log-in and registration pages, and the history page. Figma was used to create wireframes and interactive prototypes, simplifying the design process and validating the layouts before development [20].

2.6.3 Development

The mobile application was developed using Flutter to create a cross-platform user interface for Android and iOS, offering high efficiency and performance [21]. The front-end was implemented in Flutter and Dart, using the Provider package for state management. Users could upload facial images through the camera or gallery for live classification. The backend used Firebase for user authentication and for storing the classification history in

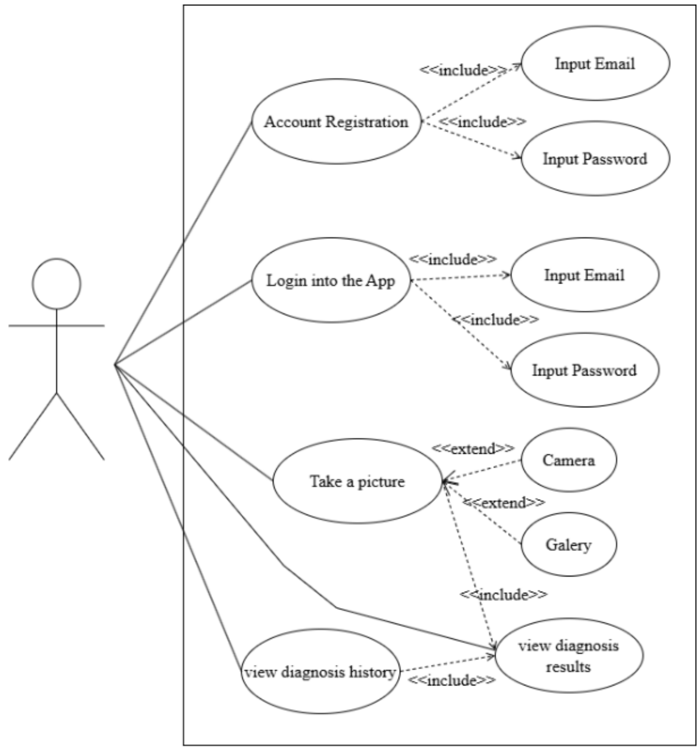

Figure 6: Use case diagram for the autism screening mobile application.

Firestore, which supports real-time synchronization and scalability [22]. The trained TFLite model was integrated to perform local inference directly on the device.

2.6.4 Testing

Functional tests were performed to ensure that all core features, including user login, image input, classification, and result history, operated correctly. This testing was crucial to maintain the quality of the application, especially in systems using real-time integration and cloud services such as Firebase [23]. The classification performance of the TFLite-integrated model was also tested on multiple mobile devices to confirm its accuracy, responsiveness, and reliability during local execution [24].

3 Results

3.1 Training And Validation Result

During the early training stages, the model exhibited signs of overfitting. Training accuracy increased from 63.23% in epoch 1 to approximately 80% in epoch 15, while validation accuracy fluctuated between 43.75% and 57.81%. After Epoch 15, performance began to stabilize. By epoch 200, the model achieved a final training accuracy of 97.63%. The validation accuracy peaked at 98.44% at epoch 187 and then settled at 93.75% by the final epoch. The loss values also showed steady improvement, with the training loss decreasing from 0.6845 to 0.1051, and the validation loss reaching its lowest value of 0.1413 in epoch 199. The model was trained using the SGD optimizer with a learning rate of 0.0001 and a batch size of 16, using binary cross-entropy as the loss function. As illustrated in Figure 7, the training and validation curves demonstrate consistent convergence over the course of 200 epochs.

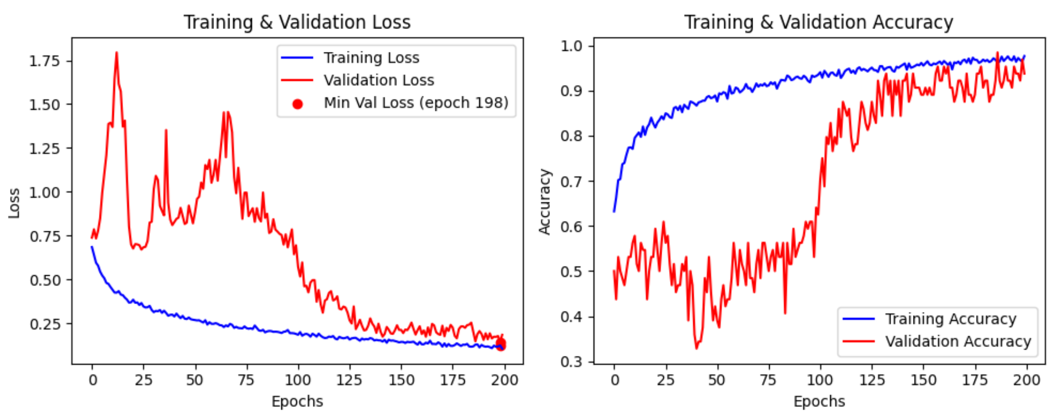

Figure 7: Accuracy and loss curves for training and validation.

These results indicate that the MobileNetV3-Small model was effectively optimized, with minimal overfitting in the later stages of training.

3.2 Evaluation Metrics

The trained model achieved robust classification performance on the test set. As shown in Figure 8, the confusion matrix highlights that out of 238 test samples, the model correctly identified 116 autistic cases and 115 nonautistic cases. Only 3 autistic samples were misclassified as non-autistic (false negatives), and 4 non-autistic samples were misclassified as autistic (false positives).

Evaluation metrics are summarized in Table 3, where both autistic and nonautistic classes achieved 0.97 in precision, recall, and F1 Score. The overall accuracy also reached 97%.

Table 3: Precision, recall, and F1-Score for autistic and non-autistic classification

Class	**Precision**	**Recall**	**F1-Score**
Autistic	0.97	0.97	0.97
Non-Autistic	0.97	0.97	0.97
Accuracy	-	-	0.97

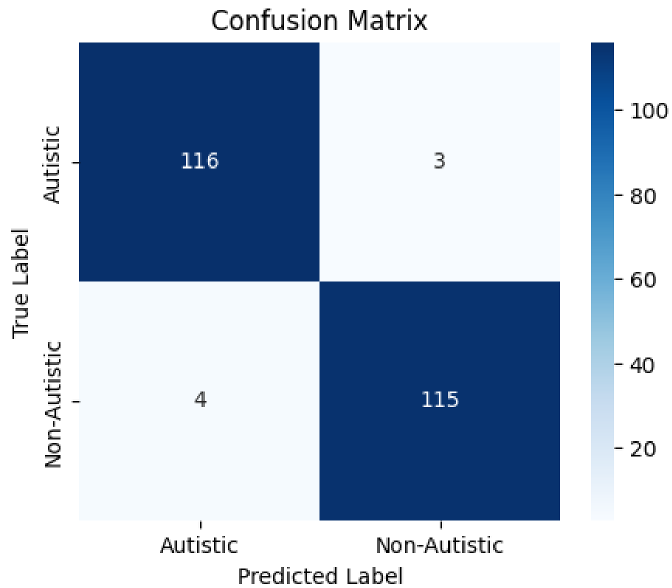

Figure 8: Classification model confusion matrix.

These results suggest a strong generalization to unseen data, with a balanced performance between classes. In real-world ASD screening, tools typically target sensitivity and specificity above 90% to be considered clinically useful. The high precision and recall achieved by this model indicate its potential for practical application in early ASD screening contexts.

Although the results are promising, statistical confidence intervals and significance tests were not performed in this study. Future work should include these analyses to confirm the robustness of the model in different test samples and scenarios.

Furthermore, although the data set covers a wide age range (3–14 years), the gender distribution was not balanced during training. This may introduce bias in model predictions, particularly if certain facial features are correlated with gender. Moreover, since the data were collected from a single geographic region (Banda Aceh, Indonesia), the model may exhibit limited generalizability to other populations. These issues highlight the need for more diverse and demographically balanced datasets in future research.

3.3 Mobile Application Development

Following model evaluation, a mobile application was developed using Flutter to provide a lightweight and responsive cross-platform interface. The mobile application was designed with ease of use in mind, allowing users to upload face images via camera or gallery, classify the image using the embedded model, and store the result in Firebase Cloud Firestore for record keeping. The previously converted TFLite model was integrated using the tflite_flutter plugin, enabling predictions to run entirely on the device. This approach eliminates the dependency on cloud-based servers, resulting in faster inference and improved data privacy. The main user interface and features of the application are displayed in Figure 9, including image input, prediction display, and history view functions.

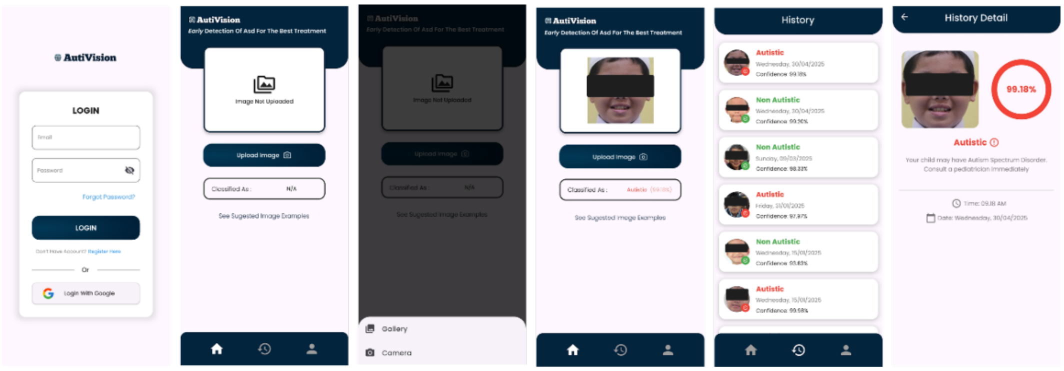

Figure 9: User interface of the autism classification application.

3.4 Application Testing

The mobile application was tested on several Android devices to evaluate its responsiveness, accuracy, and deployment performance. The classification results were displayed within 1 to 2 seconds after an image was uploaded, accompanied by a confidence score. After classification, the results were stored in the Firebase Firestore, with the upload speed being dependent on the quality of the device's network. The application demonstrated smooth operation with minimal latency across various devices, validating the feasibility of real-time on-device inference.

To assess its efficiency, the proposed model was compared with the architectures used in related studies. For example, ResNet-50, often used in research related to ASD, contains more than 25 million parameters and requires approximately 4.1 GFLOPs for inference. MobileNetV2 is lighter, but still uses about 3.4 million parameters and 300 MFLOPs. In contrast, MobileNetV3-Small, as implemented in this study, contains only 976,396 parameters and requires approximately 116.08 MFLOPs, making it significantly more efficient for mobile deployment.

Despite its compact size, the model maintained a high classification accuracy of 97%, highlighting that it is suitable for use in real time in mobile environments, where computational efficiency, speed, and energy consumption are critical.

4 Discussion

The proposed MobileNetV3-Small model demonstrated strong performance in classifying facial images of autistic and non-autistic children, achieving a precision, recall, and F1 Score of 0.97 on the test dataset. With only 976,396 parameters and 116.08 MFLOPs, the model is significantly lighter than architectures such as ResNet-50, allowing for fast inference (1 to 2 seconds) on mobile devices. This combination of accuracy and efficiency makes it particularly suitable for real-time screening applications in low-resource environments.

The successful integration of the TensorFlow Lite model into a Flutter-based mobile application resulted in a responsive and user-friendly system. The use of Firebase for history storage further enhances usability for both individuals and practitioners who may require record-keeping and follow-up analysis.

Despite the promising results, several limitations remain. The dataset used was relatively small and demographically limited, with all images collected from schools in Banda Aceh, Indonesia. Furthermore, the gender distribution was not balanced and metadata such as ethnicity was not documented. These factors raise concerns about the fairness and generalizability of the model.

Although unseen test images were used, no statistical confidence intervals or significance tests were performed to verify robustness. Future work should address this by evaluating the model using diverse publicly available benchmark datasets and applying statistical tests such as bootstrapped confidence intervals. It is also important to assess fairness across subgroups (e.g., age, gender) and test performance under varied lighting or camera conditions to simulate real-world usage. Incorporating clinician-in-the-loop feedback and adversarial robustness checks would further improve the model's clinical reliability and adoption potential.

5 Conclusion

This study successfully developed a Flutter-based mobile application to classify facial images of autistic and non-autistic children using the MobileNetV3-Small architecture, achieving 97% precision, precision, recall, and F1 Score on test data. The lightweight model, integrated via TensorFlow Lite, enabled fast, on-device inference and was deployed in a responsive interface with Firebase-based history storage. Although the results are promising, the dataset was limited to 600 images captured indoors from a single region with an unbalanced gender distribution, raising concerns about fairness and generalizability. Future work should involve multisite data collection, performance benchmarking on public datasets, testing under diverse conditions, and validation involving clinical professionals to ensure robustness and real-world applicability.

Acknowledgments

The authors express gratitude to Universitas Syiah Kuala for supporting this study.

References

[1] W. Jones, C. Klaiman, S. Richardson, M. Lambha, M. Reid, T. Hamner, C. Beacham, P. Lewis, J. Paredes, L. Edwards, N. Marrus, J. N. Constantino, S. Shultz, and A. Klin, "Development and Replication of Objective Measurements of Social Visual Engagement to Aid in Early Diagnosis and Assessment of Autism," *JAMA Network Open*, vol. 6, p. e2330145, Sept. 2023.

[2] J. Zeidan, E. Fombonne, J. Scorah, A. Ibrahim, M. S. Durkin, S. Saxena, A. Yusuf, A. Shih, and M. Elsabbagh, "Global prevalence of autism: A systematic review update," *Autism Research*, vol. 15, pp. 778–790, May 2022.

[3] Z. A. T. Ahmed, T. H. H. Aldhyani, M. E. Jadhav, M. Y. Alzahrani, M. E. Alzahrani, M. M. Althobaiti, F. Alassery, A. Alshaflut, N. M. Alzahrani, and A. M. Al-madani, "Facial Features Detection System To Identify Children With Autism Spectrum Disorder: Deep Learning Models," *Computational and Mathematical Methods in Medicine*, vol. 2022, pp. 1–9, Apr. 2022.

[4] M. S. Alam, M. M. Rashid, A. R. Faizabadi, H. F. Mohd Zaki, T. E. Alam, M. S. Ali, K. D. Gupta, and M. M. Ahsan, "Efficient Deep Learning-Based Data-Centric Approach for Autism Spectrum Disorder Diagnosis from Facial Images Using Explainable AI," *Technologies*, vol. 11, p. 115, Aug. 2023.

[5] M. G. Perinelli and M. Cloherty, "Identification of autism in cognitively able adults with epilepsy: A narrative review and discussion of available screening and diagnostic tools," *Seizure: European Journal of Epilepsy*, vol. 104, pp. 6–11, Jan. 2023.

[6] N. Arora, P. Yadav, K. Tripathi, and S. Sharma, "Performance of CNN for different facial expression images with varying input dataset sizes," in *2023 International Conference on Device Intelligence, Computing and Communication Technologies, (DICCT)*, (Dehradun, India), pp. 351–355, IEEE, Mar. 2023.

[7] J. ElMahalawy, Y. A. ElSwaify, D. Elliboudy, O. M. Abbas, N. Moustafa, and N. Wael, "AI-Powered Human-Computer Interaction Assisting Early Identification of Emotional and Facial Symptoms of Autism Spectrum Disorder in Children: "A Deep Learning-Based Enhanced Facial Feature Recognition System"," in *2024 International Conference on Machine Intelligence and Smart Innovation (ICMISI)*, (Alexandria, Egypt), pp. 87–93, IEEE, May 2024.

[8] F. Nowak, K.-W. Yung, J. Sivaraj, P. De Coppi, D. Stoyanov, S. Loukogeorgakis, and E. B. Mazomenos, "An investigation into augmentation and preprocessing for optimising X-ray classification in limited datasets: a case study on necrotising enterocolitis," *International Journal of Computer Assisted Radiology and Surgery*, vol. 19, pp. 1223–1231, Apr. 2024.

[9] Z. S. Abu Ghrban and N. K. E. Abbadi, "Human Age Predication from Face Images Based on Combining Deep Wavelet Network and Machine Learning Algorithms," *Journal of Computer Science*, vol. 19, pp. 654–666, May 2023.

[10] K. Goel, H. Dhiman, H. Yadav, and S. Kumar, "Facial Image Pre-processing for Enhanced Ethnicity Classification," in *2023 3rd International Conference on Intelligent Technologies (CONIT)*, (Hubli, India), pp. 1–6, IEEE, June 2023.

[11] P. Sengupta, A. Mehta, and P. S. Rana, "Enhancing Performance of Deep Learning Models with a Novel Data Augmentation Approach," in *2023 14th International Conference on Computing Communication and Networking Technologies (ICCCNT)*, (Delhi, India), pp. 1–7, IEEE, July 2023.

[12] Herimanto, Arie Satia Dharma, Junita Amalia, David Largo, and Christin Adelia Pratiwi Sihite, "Performance Analysis of MobileNetV3-based Convolutional Neural Network for Facial Skin Disorder Classification," *Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi)*, vol. 8, pp. 701–709, Dec. 2024.

[13] J. Pathmarasa, U. Wijenayake, R. E. Wijesinghe, and B. N. Silva, "An Exploratory Study of Diverse Models and Datasets for Transfer Learning based Image Classification on Sparse Data," in *2024 Moratuwa Engineering Research Conference (MERCon)*, (Moratuwa, Sri Lanka), pp. 324–329, IEEE, Aug. 2024.

[14] V. Y. Bhole and A. Kumar, "Flutter-Based Real-Time Mobile App for Fruit Shelf-Life Prediction (FSP) Using Multi-Modality Imaging:," *International Journal of Information Retrieval Research*, vol. 12, pp. 1–20, Apr. 2022.

[15] D. Krstinić, A. K. Skelin, I. Slapničar, and M. Braović, "Multi-Label Confusion Tensor," *IEEE Access*, vol. 12, pp. 9860–9870, 2024.

[16] O. Mohammed Ahmed, "Backend as a Service Cloud Computing Integrated with Cross-platform Mobile Development Framework to Create an E-learning application that works in Mobile and Web with a single codebase," in *4th International Conference on Communication Engineering and Computer Science (CIC-COCOS'2022)*, pp. 50–57, Cihan University, 2022.

[17] M. Muneeb, S. F. Feng, and A. Henschel, "Deep Learning Pipeline for Image Classification on Mobile Phones," in *Artificial Intelligence and Applications*, pp. 1–20, Academy and Industry Research Collaboration Center (AIRCC), May 2022.

[18] W. K. A. Alabayechi and S. M. Hadi, "Developing a Network-Based Android Messaging Application," in *2023 International Conference on Engineering Applied and Nano Sciences (ICEANS)*, (Erbil, Iraq), pp. 44–48, IEEE, Oct. 2023.

[19] Yoyon Efendi, Ilham Perdana, Muhammad Raihan, Rometdo Muzawi, Nurul Utami, and Tashid, "IOS-Based Mobile Service Ordering Application Using Wireframe and Figma," *JAIA - Journal of Artificial Intelligence and Applications*, vol. 2, pp. 1–7, Apr. 2022.

[20] G. Goel, P. Tanwar, and S. Sharma, "UI-UX Design Using User Centred Design (UCD) Method," in *2022 International Conference on Computer Communication and Informatics (ICCCI)*, (Coimbatore, India), pp. 1–8, IEEE, Jan. 2022.

[21] L. Heimann and O. Veliz, "Mobile Application Development in Flutter," in *Proceedings of the 53rd ACM Technical Symposium on Computer Science Education V. 2*, (Providence RI USA), pp. 1199–1199, ACM, Mar. 2022.

[22] K. Marimuthu, A. Panneerselvam, S. Selvaraj, L. P. Venkatesan, and V. Sivaganesan, "Android Based College App Using Flutter Dart," *Green Intelligent Systems and Applications*, vol. 3, pp. 69–85, Aug. 2023.

[23] Jadhav Prashant, Bhujbal Pranav, Kapkar Arpit, Gayke Krushna, and Prof. Mrs. S. S. Patil, "EvPowerStop - Electric Vehicle Station Finder and Slot Scheduler," *International Journal of Advanced Research in Science, Communication and Technology*, pp. 366–369, Oct. 2024.

[24] M. Krumnikl and V. Maiwald, "Facial Emotion Recognition for Mobile Devices: A Practical Review," *IEEE Access*, vol. 12, pp. 15735–15747, 2024.