

# Classification of Autism Spectrum Disorder Based on Facial Images Using the VGG19 Algorithm

Wandi Yusuf Kurniawan and Putu Harry Gunawan\*

Human Centric Engineering (HUMIC), School of Computing, Telkom University, Bandung, Indonesia  
onedemif@student.telkomuniversity.ac.id, phgunawan@telkomuniversity.ac.id

## Abstract

Autism spectrum disorder profoundly affects early communication and physical skills, emphasizing the need for effective interventions. Approximately one in a hundred children worldwide is affected by autism. The convolutional neural network (CNN), especially VGG19, is the most accurate tool for detecting autism using a facial image dataset. Notably, there are many configurations that can be applied to produce the best accuracy. This study evaluated how facial images can be used to classify autism using VGG19-based deep learning models with different configurations, such as long-short term memory (LSTM) and Dropout layers; adaptive moment estimation (Adam), root mean square propagation (RMSprop), and stochastic gradient descent (SGD) optimizers; and a cosine annealing learning rate scheduler. Results highlighted substantial performance variations across the configurations, with RMSprop+LSTM+Dropout achieving the highest accuracy (75.85%), average precision, non-autistic precision, and average F1-score. Notably, Adam showed the best performance in non-autistic precision (83.09%) and autistic F1-score (76.74%), while Adam+LSTM+Dropout demonstrated superior autistic precision (85.16%) and non-autistic recall (90.82%). Moreover, SGD+Dropout achieved the highest autistic recall (91.84%). Selecting an appropriate configuration is crucial, and further research can help optimize the architecture, activation functions, and preprocessing for enhanced accuracy. High-accuracy models hold promise for aiding autism detection and communication and physical skill development.

**Category:** Computer Graphics / Image Processing

**Keywords:** Autism spectrum disorder; Facial image; CNN-VGG19; Classification; Optimizer

## I. INTRODUCTION

Autism spectrum disorder, or autism, is a neurological disorder that has been shown to have a significant impact on children's communication and interaction abilities, such as eye responsiveness and social interaction, at an early age [1]. Individuals with 10-item Autism-Spectrum Quotient (AQ10) show less participation in physical activities compared to those with typical development [2], which can result in serious consequences, such as

depression and suicidal thoughts [3]. Therefore, it is crucial to provide effective therapy and assistance to individuals who have autism.

Zeidan et al. [4] reported that around one in a hundred children worldwide is affected by autism, with a male-to-female ratio of 21:5. The prevalence of autism was systematically examined and factors contributing to variability in estimates were considered. Changes in prevalence over time and variations within sociodemographic groups were noted, reflecting the evolving definition of

**Open Access** <http://dx.doi.org/10.5626/JCSE.2024.18.1.1>

<http://jcse.kiise.org>

This is an Open Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License (<http://creativecommons.org/licenses/by-nc/4.0/>) which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.

Received 19 February 2024; Accepted 12 March 2024

\*Corresponding Author

autism and differences in the methodology and contexts of prevalence studies.

In the development of facial recognition methods using images and convolutional neural networks (CNN), especially those developed by the Keras library, such as the Visual Geometry Group with nineteen layers (VGG19), preprocessing stages such as resolution standardization, color or feature extraction, and separation into training and testing data are carried out. This approach has been proven to have higher accuracy than other machine learning methods [5, 6]. Researchers have also improved the performance of this model by using transfer learning methods. Ghazal et al. [7] reported a satisfactory accuracy rate of 87.7% and Sadik et al. [8] demonstrated a 6% improvement when employing transfer learning.

The long-short term memory (LSTM) neural network, a type of recurrent neural network (RNN), is well-known for its ability to analyze sequences of data in different fields. It's used to detect arrhythmias [9], transcribe speech into text [10], and identify coronavirus disease 2019 from X-ray images [11]. However, its application to directly identify autism through facial recognition has been limited, as observed in recent research by Saranya and Anandan [12].

In the context of deep learning, adding dropout layers as a regularization technique has consistently been shown to improve training and validation performance, especially when working with difficult datasets such as that of the Canadian Institute for Advanced Research with ten and one hundred different classes (CIFAR-10 and CIFAR-100). For example, Inoue [13] significantly reduced error rates by implementing dropout layers in their research. These findings were similar to those of Lee and Lee [14], who found that using a dropout layer with a probability of 0.05 in the VGG19 architecture cut the error rate by 0.17% for CIFAR-10 and 0.24% for CIFAR-100.

Moreover, it's crucial to note the plethora of algorithms available in deep learning, such as adaptive moment estimation (Adam), root mean square propagation (RMSprop), and stochastic gradient descent (SGD) [15]. The choice of learning rate scheduling, like the cosine annealing

approach, can significantly impact model performance. Improper learning rates, whether too high or too low, can hinder the convergence process [16].

In this study, the VGG19-based architecture for transfer learning uses LSTM and dropout layers; Adam, RMSprop, and SGD optimizer algorithms; and cosine annealing-based learning rate scheduling. The main goal was to explore the performance of this architecture using various configurations in recognizing autism and non-autism classes in facial image datasets.

This paper is divided into several sections. The next section will provide an overview of the Kaggle dataset, introduce the proposed modeling framework, and explain the evaluation criteria used. The third section presents experimental results, performs analytical comparisons against deep learning frameworks, and highlights the advantages of this approach. The final section serves as a conclusion and suggests future research directions.

## II. METHODOLOGY

The dataset was clustered using the splitting technique. The procedure for classifying autism using facial images is illustrated in Fig. 1.

- Dataset preparation: The process begins with collecting and organizing the dataset containing facial images.
- Split, resize, and extract the dataset: The dataset is divided into training and testing data. The images are resized to a standard format and are preprocessed.
- Training data and testing data: The split dataset is divided into two sets: training the model and evaluating its performance.
- Modifying layers, compilers, and learning rate: Adjustments are made to the layers of the VGG19 model, compilation settings are defined, and a learning rate schedule is established.
- VGG19 model preparation: The VGG19 model is prepared for training, considering the modifications and settings made in the previous step.

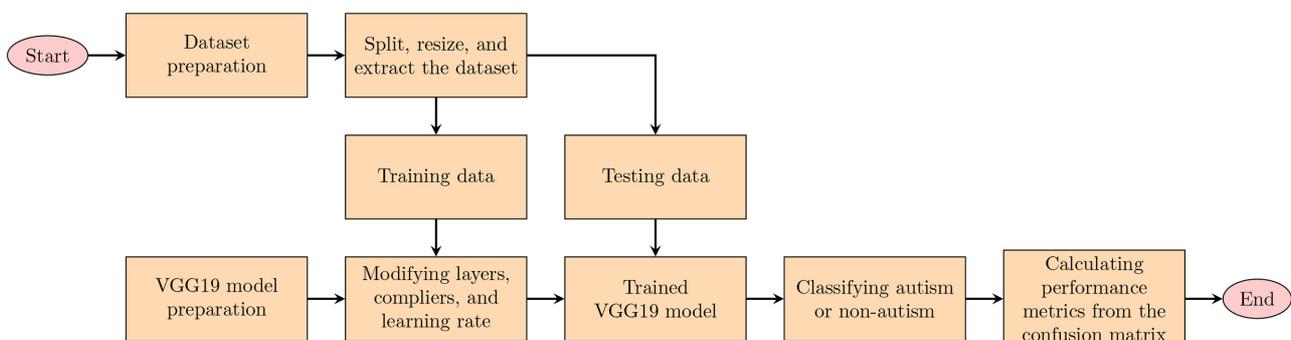


Fig. 1. Scenarios in autism classification with facial images.

- Trained VGG19 model: The model is trained on the training data to learn patterns and features of the facial images.
- Classifying autism or non-autism: The trained model is used to classify facial images into the autistic or non-autistic category.
- Calculating performance metrics from the confusion matrix: The model’s performance is assessed by calculating various metrics, such as accuracy, precision, recall, and F1-score, based on the confusion matrix.

### A. Autism Image Dataset

According to Aldridge et al. [17], autistic patients have unique facial expressions or characteristics. The characteristics that distinguish autistic patients are the shape of the forehead, eyes, philtrum (indentation between the nose and lips), and a wider mouth. This makes the faces of autistic patients shorter than those of other people. Visual examples of autistic and normal patients are shown in Fig. 2.

The dataset utilized in this study consisted of 2,940 facial images of children aged 2–14 years old, sourced from various online platforms and available on Kaggle (<https://www.kaggle.com/datasets/cihan063/autism-image-data>). These images were categorized into autistic and non-autistic facial images, each containing 1,470 images.

Subsequently, the dataset was divided into 80% training

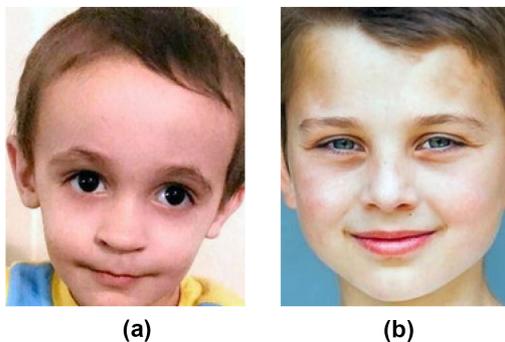


Fig. 2. Example facial images depicting autism (a) and non-autism (b).

Train			Valid 1	Test
Train		Valid 2	Train	Test
Train		Valid 3	Train	
Train	Valid 4	Train		Test
Valid 5	Train			Test

Fig. 3. Representation of dataset division into training (blue), validation (red), and testing (green) set across five-folds, with numbers denoting the respective validation fold.

and 20% testing data. Within the training data, one of every five segments, referred to as “folds,” was utilized as validation data, as demonstrated in Fig. 3. Following this, all data segments underwent processing using the OpenCV library to adjust their resolution to 224×224 pixels, resulting in color extraction in the red, green, and blue channels. Each pixel was assigned a value ranging from 0 to 255.

### B. VGG19 Model and Its Configurations

In this study, the VGG19 model was used to predict whether facial images belonged to autistic or non-autistic people by using weights from ImageNet without changing them to maintain accuracy and loss values [18, 19]. Comprehensive visualization of the fundamental architecture can be seen in Fig. 4.

The model was modified by substitution after the fifth MaxPooling2D layer, situated in the fifth green box, with a Reshape and LSTM layer. The LSTM utilized a hyperbolic tangent activation function [20, 21], subsequently undergoing normalization and flattening. After the LSTM layer, a dense layer with a rectified linear unit (ReLU) activation function was added. This activation function has been shown to work well and consistently across different neural networks [22].

A dropout layer, positioned after the dense layer, randomly deactivated a specific number of neurons with a designated probability to mitigate the risk of overfitting. Finally, a single classification output was made using a sigmoid activation function, which worked better than the softmax function, usually used for multiclass classification [23].

After the model architecture was built and trained on the training data, the binary cross-entropy function was used to determine the extent of loss [24]. Three optimizers, for optimizing the model, namely Adam, RMSprop, and SGD, were tested [15]. Additionally, a cosine annealing learning rate scheduler was applied to all configurations. The equation used for this scheduling was as follows:

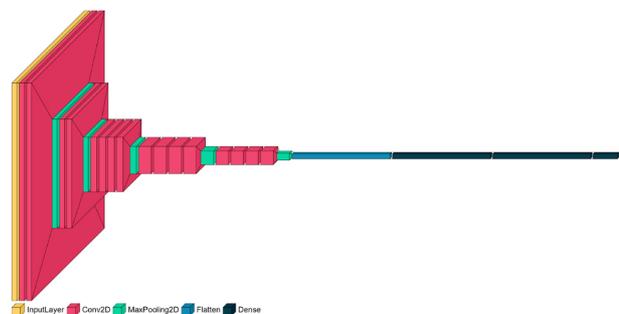
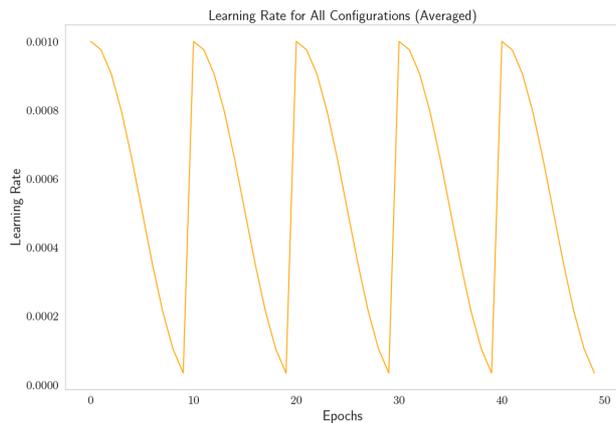


Fig. 4. VGG19’s full model architecture.



**Fig. 5.** Cosine annealing learning rate scheduler for all configurations.

$$\eta_t = \eta_{min} + \frac{1}{2}(\eta_{max} - \eta_{min})\left(1 + \cos\left(\frac{t}{T}\pi\right)\right), \quad (1)$$

where  $\eta_t$  represents the current learning rate value since the last restart,  $\eta_{min} = 10^{-5}$  denotes the minimum achievable learning rate,  $\eta_{max} = 10^{-3}$  is initially set as the learning rate value,  $t$  denotes the number of epochs elapsed since the last restart, and  $T = 10$  signifies the maximum number of epochs for each fold. This learning rate was applied in five cycles for all configurations, corresponding to the number of folds, as illustrated in Fig. 5.

Twelve configurations were utilized in this study, each representing a different combination of optimizers and adding different layers. A comprehensive exploration of how the VGG19 model could be employed to classify autism was done through the following configurations:

1. Adam optimizer with no layer addition.
2. Adam optimizer with LSTM layer addition.
3. Adam optimizer with Dropout layer addition.
4. Adam optimizer with LSTM and Dropout layer addition.
5. RMSprop optimizer with no layer addition.
6. RMSprop optimizer with LSTM layer addition.
7. RMSprop optimizer with Dropout layer addition.
8. RMSprop optimizer with LSTM and Dropout layer addition.
9. SGD optimizer with no layer addition.
10. SGD optimizer with LSTM layer addition.
11. SGD optimizer with Dropout layer addition.
12. SGD optimizer with LSTM and Dropout layer addition.

### C. Performance Metrics

The calculation of performance metrics was done using the confusion matrix and is shown in Fig. 6.

The confusion matrix is a tool used to assess the

		Predicted	
		Autistic	Non-Autistic
Actual	Autistic	TP	FN
	Non-Autistic	FP	TN

**Fig. 6.** Confusion matrix for classifying autistic and non-autistic facial images.

performance of a model that classifies individuals as either autistic or non-autistic based on certain criteria, such as facial features and it relates to the context of autism detection in the following ways:

- True positive (TP): This refers to the number of cases where the model correctly predicts an individual as autistic, and indeed, that person is autistic. In autism detection, a TP represents a successful identification of an autistic individual.
- True negative (TN): This indicates the number of cases where the model correctly predicts an individual as autistic, and that person is indeed non-autistic. In this context, a TN means the model accurately identifies someone as not having autism.
- False positive (FP): This describes the instances where the model incorrectly predicts an individual as autistic, but that person is non-autistic. In the context of autism detection, an FP means the model mistakenly identified a non-autistic individual as autistic.
- False negative (FN): This refers to the cases where the model incorrectly predicts an individual as non-autistic, but that person is autistic. In the context of autism detection, an FN means the model failed to identify someone with autism.

After obtaining information about the confusion matrix for each configuration, various evaluation metrics are used to assess the performance of the VGG19 model. These metrics include accuracy, precision, recall, and F1-score [25]. These metrics were calculated using the equation below:

$$\text{Accuracy} = \frac{TP+TN}{TP+FP+FN+TN}, \quad (2)$$

$$\text{Precision} = \frac{TP}{TP+FP}, \quad (3)$$

$$\text{Recall} = \frac{TP}{TP+FN}, \tag{4}$$

$$F1\text{-score} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}. \tag{5}$$

### III. RESULTS AND DISCUSSION

#### A. Training and Validation

The training, validating, and testing of the VGG19 model for all configurations on Kaggle’s graphics processing unit (GPU) took approximately 30 minutes. Five-fold cross-validation, with ten epochs per fold, was employed for each configuration. The training and validation outcomes are illustrated in Figs. 7 and 8.

Fig. 7 illustrates the accuracy distribution during training and validation across all model configurations. The median accuracy reflects the model’s predictive performance, while the accuracy range indicates the consistency of these predictions. Results reveal that the Adam+LSTM configuration achieved the highest median training accuracy, while SGD+LSTM had the lowest median. During validation, most configurations exhibited higher medians, wider ranges, and the presence of outliers below

the range threshold, indicating increased bias and variability compared to the training phase. Significant variation in the validation accuracy range was observed, with RMSprop+LSTM+Dropout having the widest range and SGD+LSTM+Dropout having the narrowest range.

Fig. 8 shows how the loss values changed during training and validation across all the configurations, excluding those that were outliers. This is especially true during the first epoch when the model was still familiarizing with the dataset so that it would be easier to analyze. The Adam+LSTM configuration had the lowest median training loss, while SGD+Dropout had the lowest median. In the validation phase, median fluctuations and wider loss ranges were observed across all configurations compared to the training phase. This suggests that the relationship between accuracy and loss can fluctuate, with high accuracy not always corresponding to low loss. RMSprop+LSTM+Dropout had the widest validation loss range, SGD+LSTM+Dropout had the lowest median validation loss range, and SGD+LSTM had the narrowest validation loss range.

#### B. Testing

A rounding technique was used to discretize test data results, requiring a consistent prediction from at least three out of five folds, regardless of the original class. An

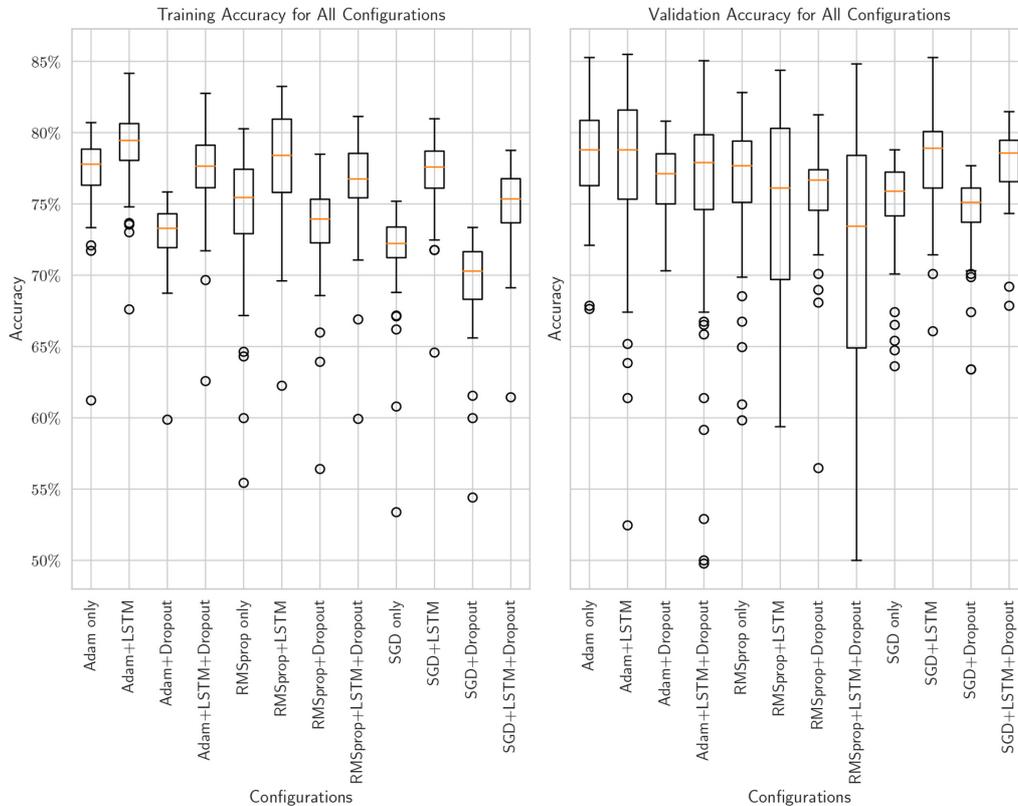
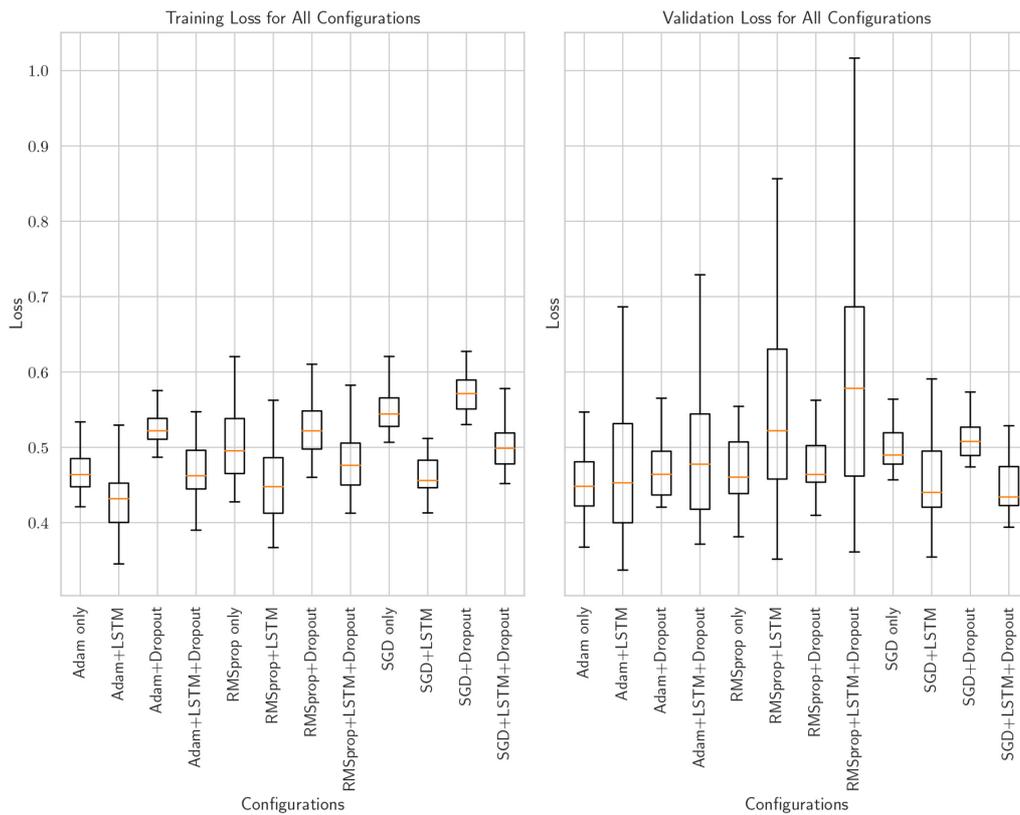


Fig. 7. Boxplots illustrating statistical descriptions of training and validation performance based on accuracy for each configuration.



**Fig. 8.** Boxplots demonstrating statistical descriptions of training and validation performance based on loss, excluding outliers.

example is shown in Fig. 9.

As illustrated in Fig. 10, the RMSprop+LSTM+Dropout configuration achieved the highest accuracy (75.85%) among all the configurations. It accurately predicted 446 facial images in the two classes, with 57 FPs and 85 FNs. In contrast, the SGD+LSTM+Dropout configuration achieved a lower accuracy (54.08%), accurately predicting 318 facial images in the two classes with 18 FPs and 182 FNs (See Table 1 for comprehensive metric results across all configurations). The accuracy in the RMSprop+LSTM+Dropout configuration is shown consistent with the result in [26].

As shown in Table 1, the RMSprop+LSTM+Dropout



First Fold: 0 (Non-autistic)  
 Second Fold: 0  
 Third Fold: 0  
 Fourth Fold: 1 (Autistic)  
 Fifth Fold: 1  
 Overall Results:  
 0 [3 out of 5 folds]

**Fig. 9.** Instances where facial images of individuals with autism were misclassified as non-autistic across five folds.

configuration exhibited the best average precision, non-autistic precision, and F1-score, at 76.09%, 76.95%, and 75.80%, respectively. Additionally, the Adam configuration achieved the highest F1-scores for non-autistic and autistic precision, at 83.09% and 76.74%, respectively. Furthermore, the Adam+LSTM+Dropout configuration demonstrated the best autistic precision (85.16%) and non-autistic recall (90.82%). Lastly, the SGD+Dropout configuration achieved the highest autistic recall (91.84%). It is important to note that each configuration possesses unique characteristics and capabilities that contributed to these results. These findings emphasize the importance of selecting the appropriate configuration based on specific classification requirements and objectives.

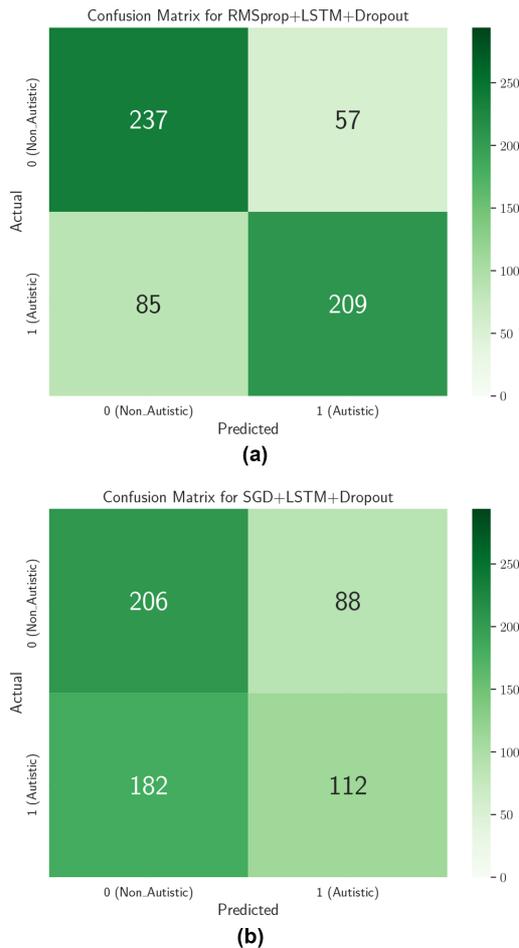
Based on the results of this study, there are opportunities for further research and development to improve model performance. Some avenues for exploration include:

- Combining publicly available datasets with in-house datasets;
- Extracting additional information from images, such as color, facial landmarks, or attributes like gender, age, race, and emotion, using OpenCV and DeepFace libraries [27];
- Exploring modifications to CNN model architectures and compilers, whether using the Keras library custom

**Table 1.** Configurations based on confusion matrix

	Accuracy (%)	Precision (%)			Recall (%)		F1-score (%)		
		Non-autistic	Autistic	Avg.	Non-autistic	Autistic	Non-autistic	Autistic	Avg.
Adam only	73.30	<b>83.09</b>	67.98	75.54	58.50	88.10	68.66	<b>76.74</b>	72.70
Adam+LSTM	70.41	67.24	75.00	71.12	79.59	61.22	72.90	67.42	70.16
Adam+Dropout	72.96	82.93	67.62	75.28	57.82	88.10	68.14	76.51	72.33
Adam+LSTM+Dropout	71.77	65.76	<b>85.16</b>	75.46	<b>90.82</b>	52.72	76.29	65.13	70.71
RMSprop only	71.43	81.82	66.15	73.99	55.10	87.76	65.85	75.44	70.65
RMSprop+LSTM	72.62	67.64	81.52	74.58	86.73	58.50	76.01	68.12	72.06
RMSprop+Dropout	74.49	78.12	71.69	74.91	68.03	80.95	72.73	76.04	74.38
RMSprop+LSTM+Dropout	<b>75.85</b>	73.60	78.57	<b>76.09</b>	80.61	71.09	<b>76.95</b>	74.64	<b>75.80</b>
SGD only	65.31	66.19	64.52	65.35	62.59	68.03	64.34	66.23	65.28
SGD+LSTM	61.22	62.60	60.12	61.36	55.78	66.67	58.99	63.23	61.11
SGD+Dropout	59.86	77.36	56.02	66.69	27.89	<b>91.84</b>	41.00	69.59	55.29
SGD+LSTM+Dropout	54.08	53.09	56.00	54.55	70.07	38.10	60.41	45.34	52.88

The bold font indicates the best average performance in each test.


**Fig. 10.** Best (a) and worst (b) confusion matrix's configuration.

designs or

- Searching for optimal parameters with GridSearchCV.

## IV. CONCLUSION

This study evaluated VGG19-based deep learning models for autism classification using facial images. Twelve configurations were assessed, each combining LSTM and Dropout layers with Adam, RMSprop, and SGD optimizers. The RMSprop+LSTM+Dropout configuration was the most successful, achieving an accuracy of 75.85%. The Adam configuration showed the best performance in non-autistic precision and autistic F1-score, while the Adam+LSTM+Dropout configuration exhibited superior autistic precision and recall. The SGD+Dropout configuration achieved the highest autistic recall. These findings emphasize the importance of selecting the appropriate model configuration based on specific classification requirements and objectives. Future research should focus on optimizing model architecture, compiler parameters, activation functions, and techniques like GridSearchCV to improve classification accuracy. Developing high-accuracy models could support parents and educators in facilitating effective communication and physical skill development in autistic children.

## CONFLICT OF INTEREST

The authors have declared that no competing interests exist.

## REFERENCES

1. S. Jahanara and S. Padmanabhan, "Detecting autism from facial image," *International Journal of Advance Research, Ideas and Innovations in Technology*, vol. 7, no. 2, pp. 219-225, 2021.
2. I. Nurhidayah, M. Kamilah, and G. G. Ramdhanie, "Physical activity of children with autism spectrum disorders: A narrative review," *Holistik Jurnal Kesehatan*, vol. 15, no. 4, pp. 581-591, 2021. <https://doi.org/10.33024/hjk.v15i4.5233>
3. B. P. Y. Harumi, N. Hartini, and I. Y. Cahyanti, "Effectiveness of cognitive behavior therapy (CBT) to treat depression on autistic adults: systematic literature review," *Journal Diversita*, vol. 7, no. 2, pp. 259-266, 2021. <https://doi.org/10.31289/diversita.v7i2.5158>
4. 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, no. 5, pp. 778-790, 2022. <https://doi.org/10.1002/aur.2696>
5. D. Cetintas, T. Tuncer, and A. Cinar, "Detection of autism spectrum disorder from changing of pupil diameter using multi-modal feature fusion based hybrid CNN model," *Journal of Ambient Intelligence and Humanized Computing*, vol. 14, pp. 11273-11284, 2023. <https://doi.org/10.1007/s12652-023-04641-6>
6. T. Akter, M. H. Ali, M. I. Khan, M. S. Satu, M. J. Uddin, S. A. Alyami, S. Ali, A. K. M. Azad, and M. A. Moni, "Improved transfer-learning-based facial recognition framework to detect autistic children at an early stage," *Brain Sciences*, vol. 11, no. 6, article no. 734, 2021. <https://doi.org/10.3390/brainsci11060734>
7. T. M. Ghazal, S. Munir, S. Abbas, A. Athar, H. Alrababah, and M. A. Khan, "Early detection of autism in children using transfer learning," *Intelligent Automation & Soft Computing*, vol. 36, no. 1, pp. 11-22, 2023. <https://doi.org/10.32604/iasc.2023.030125>
8. R. Sadik, S. Anwar, and M. L. Reza, "AutismNet: recognition of autism spectrum disorder from facial expressions using MobileNet architecture," *International Journal*, vol. 10, no. 1, pp. 327-334, 2021. <https://doi.org/10.30534/ijatcse/2021/471012021>
9. C. Chen, Z. Hua, R. Zhang, G. Liu, and W. Wen, "Automated arrhythmia classification based on a combination network of CNN and LSTM," *Biomedical Signal Processing and Control*, vol. 57, article no. 101819, 2020. <https://doi.org/10.1016/j.bspc.2019.101819>
10. H. J. Song, H. K. Kim, J. D. Kim, C. Y. Park, and Y. S. Kim, "Inter-sentence segmentation of YouTube subtitles using long-short term memory (LSTM)," *Applied Sciences*, vol. 9, no. 7, article no. 1504, 2019. <https://doi.org/10.3390/app9071504>
11. M. Z. Islam, M. M. Islam, and A. Asraf, "A combined deep CNN-LSTM network for the detection of novel coronavirus (COVID-19) using X-ray images," *Informatics in Medicine Unlocked*, vol. 20, article no. 100412, 2020. <https://doi.org/10.1016/j.imu.2020.100412>
12. A. Saranya and R. Anandan, "Facial action coding and hybrid deep learning architectures for autism detection," *Intelligent Automation & Soft Computing*, vol. 33, no. 2, pp. 1167-1182, 2022. <https://doi.org/10.32604/iasc.2022.023445>
13. H. Inoue, "Multi-sample dropout for accelerated training and better generalization," 2019 [Online]. Available: <https://arxiv.org/abs/1905.09788>.
14. S. Lee and C. Lee, "Revisiting spatial dropout for regularizing convolutional neural networks," *Multimedia Tools and Applications*, vol. 79, pp. 34195-34207, 2020. <https://doi.org/10.1007/s11042-020-09054-7>
15. S. H. Haji and A. M. Abdulazeez, "Comparison of optimization techniques based on gradient descent algorithm: a review," *PalArch's Journal of Archaeology of Egypt/ Egyptology*, vol. 18, no. 4, pp. 2715-2743, 2021.
16. W. C. Cheng, H. C. Hsiao, and L. H. Li, "Deep learning mask face recognition with annealing mechanism," *Applied Sciences*, vol. 13, no. 2, article no. 732, 2023. <https://doi.org/10.3390/app13020732>
17. K. Aldridge, I. D. George, K. K. Cole, J. R. Austin, T. N. Takahashi, Y. Duan, and J. H. Miles, "Facial phenotypes in subgroups of prepubertal boys with autism spectrum disorders are correlated with clinical phenotypes," *Molecular Autism*, vol. 2, article no. 15, 2011. <https://doi.org/10.1186/2040-2392-2-15>
18. K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014 [Online]. Available: <https://arxiv.org/abs/1409.1556>.
19. O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, et al., "ImageNet large scale visual recognition challenge," *International Journal of Computer Vision*, vol. 115, pp. 211-252, 2015. <https://doi.org/10.1007/s11263-015-0816-y>
20. S. K. Roy, S. Manna, S. R. Dubey, and B. B. Chaudhuri, "LiSHT: non-parametric linearly scaled hyperbolic tangent activation function for neural networks," 2019 [Online]. Available: <https://arxiv.org/abs/1901.05894>.
21. T. De Ryck, S. Lanthaler, and S. Mishra, "On the approximation of functions by tanh neural networks," *Neural Networks*, vol. 143, pp. 732-750, 2021. <https://doi.org/10.1016/j.neunet.2021.08.015>
22. A. Maniatopoulos and N. Mitianoudis, "Learnable leaky relu (LeLeLU): an alternative accuracy-optimized activation function," *Information*, vol. 12, no. 12, article no. 513, 2013. <https://doi.org/10.3390/info12120513>
23. G. Ismail, K. Chesoli, G. Moni, and K. Gikunda, "Comparison of probabilistic deep learning methods for autism detection," 2023 [Online]. Available: <https://arxiv.org/abs/2303.12707>.
24. Y. Ho and S. Wookey, "The real-world-weight cross-entropy loss function: modeling the costs of mislabeling," *IEEE Access*, vol. 8, pp. 4806-4813, 2020. <https://doi.org/10.1109/ACCESS.2019.2962617>
25. M. Heydarian, T. E. Doyle, and R. Samavi, "MLCM: multi-label confusion matrix," *IEEE Access*, vol. 10, pp. 19083-19095, 2022. <https://doi.org/10.1109/ACCESS.2022.3151048>
26. W. Y. Kurniawan, P. H. Gunawan, and N. Aquarini, "Comparison of the Keras-LSTM algorithms for classifying autism spectrum disorder using facial images," in *Proceedings of 2023 International Conference on Data Science and Its Applications (ICoDSA)*, Bandung, Indonesia, 2023, pp. 7-12. <https://doi.org/10.1109/ICoDSA58501.2023.10276630>
27. S. I. Serengil and A. Ozpinar, "Hyperextended lightface: a facial attribute analysis framework," in *Proceedings of 2021 International Conference on Engineering and Emerging Technologies (ICEET)*, Istanbul, Turkey, 2021, pp. 1-4. <https://doi.org/10.1109/ICEET53442.2021.9659697>



**Wandi Yusuf Kurniawan** <https://orcid.org/0009-0009-6027-4250>

---

Wandi Yusuf Kurniawan is a master's student at Department of Informatics, School of Computing, Telkom University, Bandung, Indonesia. He received the B.Sc. degree in Computer Science from Telkom University, Bandung, Indonesia in 2023. He is active as a teaching and research assistant related to mathematics and data science at the School of Computing, Telkom University, Bandung, Indonesia. His research interests are in machine learning and numerical methods



**Putu Harry Gunawan** <https://orcid.org/0000-0002-3635-894X>

---

Putu Harry Gunawan is an Associate Professor at the Department of Informatics, School of Computing, Telkom University, Bandung, Indonesia. He received the B.S. in Computational Mathematics from Udayana University, Indonesia in 2009, the M.Sc. dual degree in Computational Science in 2011 from ITB, Indonesia and Kanazawa University, Japan, and the Doctoral dual degree in Applied Mathematics from ITB and Pari-Est, France in 2015. He has authored five local lecture books in numerical mathematics and computational statistics, several publications in top journals, such as Computers & Fluids, Computational Geosciences, and the Progress in Computational Fluid Dynamics (CFD). Moreover, he has authored more than 70 papers for international conferences in IEEE, Springer, Elsevier, and Journal of Physics: Conference Series. His research interests are in the field of computational science, numerical methods, modeling and simulation, and parallel scientific computing.