International Journal of Computing and Digital Systems
2024, VOL. 17, NO. 1, 1–10

http://dx.doi.org/10.12785/ijcds/XXXXXX

Autism Detection Based on Digital Facial Image using Hybrid Method of CNN, LBP, and HOG

Doni Tan Hero[1] **and Afiahayati**[2]

[1]*Master Program in Computer Science, Department of Computer Science and Electronics, Universitas Gadjah Mada, Yogyakarta, Indonesia*
[2]*Department of Computer Science and Electronics, Universitas Gadjah Mada, Yogyakarta, Indonesia*

Received xx September 2024, Revised xxx 2024, Accepted xxx 2024

Abstract: Autism Spectrum Disorder is essentially a condition that disrupts a child's development, affecting communication skills, understanding, learning, and interaction difficulties. This study aims to combine LBP (Local Binary Patterns) and HOG (Histogram of Oriented Gradients) as feature extraction methods in a CNN classifier to improve its accuracy for Autism detection based on digital facial image. The process followed in this research involves sequentially applying LBP, preprocessing, applying HOG, hyperparameter tuning, training, testing, and evaluation. The results show that combining LBP and HOG feature extraction methods yields an accuracy of 80%, which is better compared to using only one of the feature extraction methods: LBP (78%) or HOG (75%), and also better than not using feature extraction at all, which results in an accuracy of 76%. The combination of feature extraction methods, as done in other research, is less appropriate in this case, as it only achieved a maximum accuracy of 66%. Therefore, feature extraction combinations must be selected carefully to avoid making it difficult for the CNN to recognize patterns in facial images. This study concludes that applying LBP as preprocessing and HOG as feature fusion on a digital image dataset can help improve CNN accuracy for early autism detection.

Keywords: Autism Spectrum Disorder, Convolutional Neural Network (CNN), Local Binary Pattern (LBP), Histogram of Oriented Gradients (HOG)

1. INTRODUCTION

Autism (Autism Spectrum Disorder) is a developmental disorder that affects children's growth, diagnosed through visual symptoms and developmental deviations such as impaired communication abilities, lack of understanding, learning difficulties, and challenges in interaction [1]. The earlier autism can be diagnosed and inferred, the greater the chance for improvement, especially if the diagnosis is followed up with therapy or enrollment in special groups [2]. Therefore, it is crucial to recognize the signs and symptoms of autism as early as possible to provide special interventions, minimizing the risk of symptoms worsening.

Recent research has begun to explore autism detection using various datasets. The datasets used include fMRI, eye gaze direction, genetic data, screening data, retinal photos, and EEG signals. However, these types of data often require significant time and financial resources. Another easily accessible dataset is digital images. Digital images can be used for autism detection because, the faces of individuals with autism exhibit certain biomarkers that differentiate them from typical faces, such as increased distance between the corners of the eyes and decreased midface length [3][4].

Deep Learning is the most commonly used classifier when attempting to classify datasets consisting of digital images, with one of the most popular models being CNN (Convolutional Neural Network). However, facial images alone may not sufficiently highlight the biomarkers present in individuals with autism, which can impact the model's performance, particularly in terms of accuracy. Poorly captured features may prevent the model from fully learning these features.

Feature extraction is one solution to emphasize important features. The goal of feature extraction is to identify informative features that can improve efficiency in the classification process [5]. Additionally, feature extraction can help reduce the resource requirements of the data without compromising important features [6].

Research on the topic of autism detection utilizing digital images has shown promise. Digital images can be used

E-mail address: doni.tan55@gmail.com, author 2, author 3

as data for autism detection because the faces of individuals with autism exhibit certain biomarkers that differentiate them from typical faces, such as increased distance between the eyes and decreased midline facial length [3][4]. Deep learning, particularly Convolutional Neural Networks (CNNs), is the most commonly used classifier when attempting to classify datasets composed of digital images. However, digital images of faces alone may not sufficiently highlight the biomarkers present in individuals with autism. This can affect the model's performance, especially in terms of accuracy. Poorly captured features can prevent the model from fully learning these features.

Feature extraction is one solution to enhance important features. The goal of feature extraction is to find informative features that can improve efficiency in the classification process [5]. Feature extraction using Local Binary Pattern (LBP) will produce features that are easier for the model to learn. However, using only LBP for feature extraction is insufficient to capture all important features in images, and it is recommended to combine it with Histogram of Oriented Gradient (HOG) [7]. Several studies have utilized facial photo datasets to detect autism. Some classifiers that provide good accuracy include CNN [8][9], Xception [10], VGG16 [11], and MobileNet [12][13]. Classifiers such as Xception, VGG16, and MobileNet are based on CNNs, which have been further developed to have specific parameters.

Feature extraction is crucial as it can identify informative features, thereby enhancing efficiency in the classification process [5]. It can help reduce resource usage in data without compromising important features [6]. One example of feature extraction is Local Binary Pattern (LBP). In a study, LBP was used to extract features from skin cancer images in a proposed model (2SGP-W), achieving over 85% accuracy on two different datasets [14]. [15] utilized LBP as a preprocessing step before feeding it into the classifier in their proposed model, which was able to achieve an accuracy of 95.7%. This indicates that LBP can be used as a feature extraction method during the preprocessing stage before being input into the classifier. Another feature extraction method is the Histogram of Oriented Gradient (HOG). A study [6] combined HOG in the autism detection process, resulting in 92.66% accuracy, which was better than CNN without HOG or SVM combined with HOG. [16] performed HOG feature extraction, which was used in combination with the results from previous feature extraction methods before being fed into the classifier. This indicates that HOG can serve as an additional feature that is combined with previously extracted features, potentially helping to improve the model's accuracy.

This research will combine HOG and LBP feature extraction and apply the extracted data to a CNN classifier. From the outset, CNN already has a feature extraction process in its initial stages. When CNN is training, especially during feature extraction, it strives to discover new, hidden, innovative, and unique features that can be effectively used for classification [17]. Therefore, if the initial data to be extracted by CNN has already undergone prior extraction, it will facilitate CNN in learning this information, which will, in turn, help improve CNN's accuracy. LBP, acting as a preprocessing step, can provide a clear facial image by ignoring other features like lighting and color, while HOG will work to extract facial features such as the eyes, mouth, and others in the form of a feature vector. Finally, CNN will learn the patterns of the previously extracted facial features for classification.

2. METHODS

On the faces of individuals with autism, there are biomarkers that can be used to differentiate between those with and without autism. This research aims to design a machine learning model capable of determining whether a person has autism. This study utilizes CNN (Convolutional Neural Network) as the method to classify a dataset of digital images, enhanced by combined feature extraction of Local Binary Pattern and Histogram of Oriented Gradients to improve detection accuracy. We also comparing our method with the others method [18] that using LBP and HOG as a direct input. The research will be divided into four scenarios: using CNN, CNN + LBP, CNN + HOG, and CNN + LBP + HOG, and CNN+LBP+HOG (direct input).

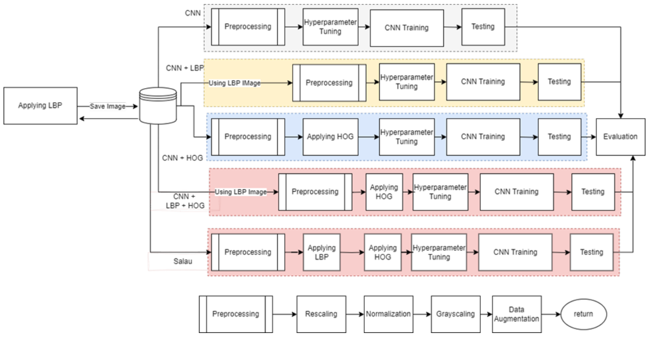

Figure 1. Research Architecture

A. Dataset

The data used comes from Kaggle [19] and is publicly available. The dataset consists of 1470 images of faces of autistic children and 1470 images of faces of non-autistic children, with varying sizes. This dataset is derived from photos collected from the internet, and the actual classes, consisting of autistic and non-autistic, were determined by the dataset owner based on their search results. However, several studies have used this dataset [20][21]. The dataset is divided into three parts: training, validation, and test, as shown in Table I. There are 1176 photos for each class (Autistic and Non-Autistic) in the training set, 144 photos for each class in the validation set, and 150 photos for each class in the test set. An artificial dataset will also be added specifically for the training data with preprocessing such as rotation, zoom, and shear. These data will be labeled with label 0 for the Autistic class and label 1 for the Non-Autistic class.

TABLE I. Dataset Description

Dataset	Composition	Percentage
Training	1176 Autistic 1176 Non-Autistic	80 %
Validation	144 Autistic 144 Non-Autistic	9,8%
Test	150 Autistic 150 Non-Autistic	10,2%

B. Preprocessing

Preprocessing will be conducted for all four scenarios. This stage is divided into four parts:

1) Resize the images to 224 x 224 pixels.
2) Normalize the pixel values from the original range of 0 to 255 to a range of 0 to 1.
3) Convert the images to Grayscale, resulting in dimensions of (224, 224, 1).
4) Apply rotation, zoom, and shear to the Artificial Dataset specifically for the training set. Rotation will vary from 0 to 180°. Zoom will range from 1.3 times the image size to 0.7 times the image size. Finally, the shear intensity will vary from 0 to 0.2.
5) Set the batch size for training and validation to 32, and for testing to 30.

C. Local Binary Pattern (LBP)

The application of Local Binary Pattern (LBP) is performed to produce a digital image in the form of a feature map that provides high-level information such as edges (high-level features), thereby reducing the workload of the CNN and simultaneously facilitating the learning process. In this study, parameters such as a radius of 1 will be used, and it will be applied to 8 points, utilizing the LBP rotation invariance type to produce images like those shown in Figure 2, Figure3, and Figure4.

D. Histogram of Oriented Gradients (HOG)

The application of Histogram of Oriented Gradients (HOG) is essentially a feature extraction method to identify important information within the image. Additionally, the application of HOG can reduce unnecessary information, thereby speeding up the extraction process. In this study, the parameters used will include a cell size of 8 x 8, a block size of 2 x 2 cells, and 9 bins consisting of 0°, 20°, 40°, 60°, 80°, 100°, 120°, 140°, 160°. The feature vector generated from this application will be used for training, validation, and testing the model.

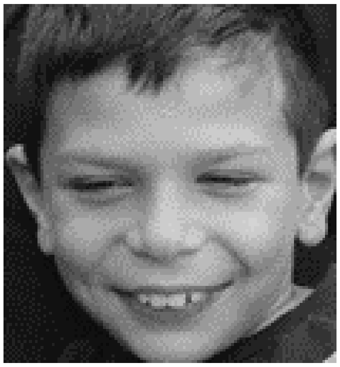

Figure 2. Normal image

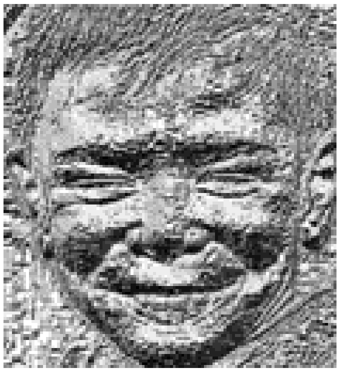

Figure 3. Default LBP

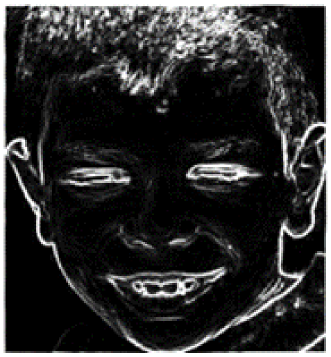

Figure 4. Rotation invariant LBP

E. Convolutional Neural Network (CNN)

This study proposes two different CNN architectures, namely CNN-1 and CNN-2, as shown in Figure 6 and

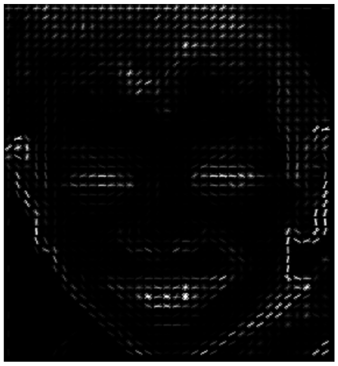

Figure 5. Results of Applying HOG

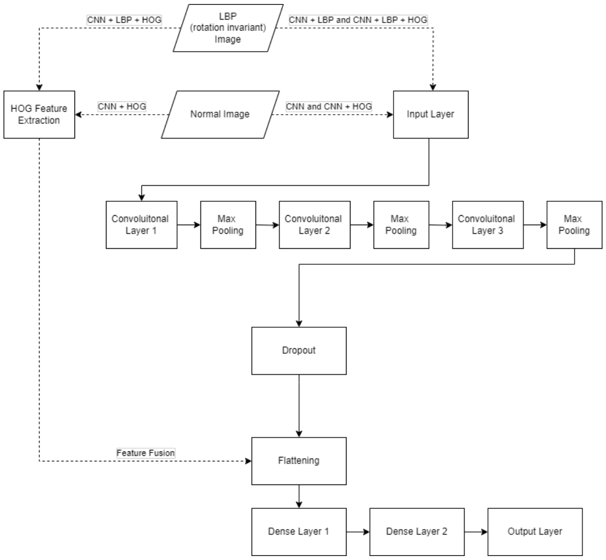

Figure 6. CNN-1 Architecture

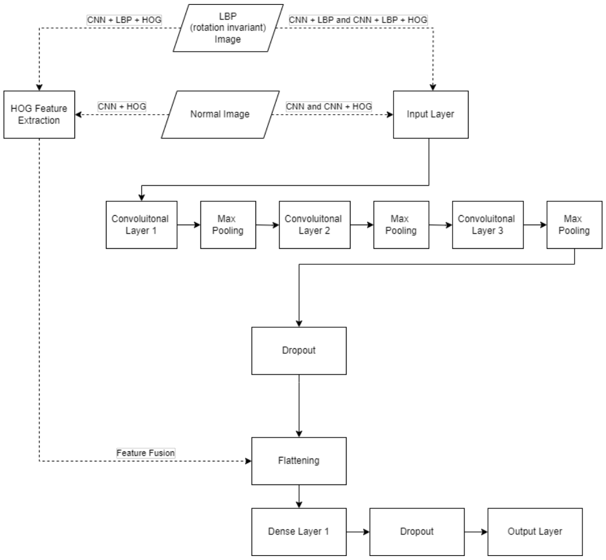

Figure 7. CNN-2 Architecture

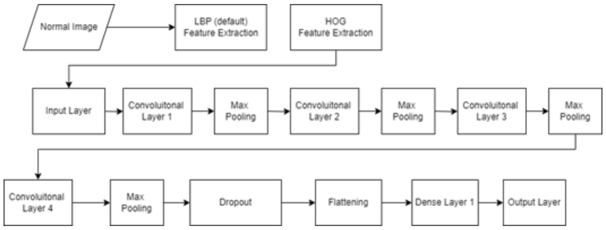

Figure 8. CNN Direct Input

Figure 7. Each architecture consists of Convolutional Layer, Max Pooling, Dropout, Flattening, Fully Connected Layer, and Output Layer. The feature fusion is performed for CNN + HOG and CNN + LBP + HOG scenario before entering the fully connected layer. The difference between the two architectures lies in the fully connected layer, where CNN-1 uses 2 dense layers, while CNN-2 uses 1 dense layer and 1 Dropout. The direct input architecture is based on a model from another study [18]. In each architecture, hyperparameter tuning will be applied to the convolutional layer, dropout, and dense layer. Each convolutional layer and dense layer will use ReLu and the dropout layer will use softmax as the activation function. The training process will run for 60 epochs, and an early stop will be applied if the model does not show improvement for the next 10 epochs.

F. Hyperparameter Tuning for CNN

This stage will involve tuning several hyperparameters in the CNN-1 and CNN-2 architectures, specifically the number of filters in the Convolutional Layer, the number of neurons in the Dense Layer, the Dropout rate, and the Learning Rate. In both CNN-1 and CNN-2 architectures, the first Convolutional Layer will apply a random value between 32 and 128, the second Convolutional Layer will apply a random value between 32 and 64, and the third or fourth Convolutional Layer will apply a random value between 16 and 32. The configuration of the Dense Layer differs between the CNN-1 and CNN-2 architectures. CNN-1 consists of two Dense Layers, where the first Dense Layer will apply a random value between 8 and 32, and the second Dense Layer will apply a random value between 16 and 64. The CNN-2 architecture has only one Dense Layer, which will apply a random value between 8 and 32. The Dropout hyperparameter in both CNN-1 and CNN-2 architectures will be selected from three possible values: 0.7, 0.6, and 0.5. Lastly, the learning rate will be set by choosing between two values: 0.0001 and 0.00001.

The tuning process will be conducted using the random search method with 10 trials. Each trial will produce a model that will then undergo training for 20 epochs, during which the best validation accuracy from all 20 epochs will be recorded. The models obtained from the 10 trials will then be ranked from the highest to the lowest validation accuracy.

G. Evaluation

At this stage, an evaluation of the resulting model will be conducted. The aspects to be evaluated are accuracy, precision, and recall for 5 scenarios. The first scenario involves not applying either LBP or HOG, the second scenario applies only LBP, the third scenario applies only HOG, and the final scenario applies both LBP and HOG. Accuracy, precision, recall, and F1-Score are used to assess the performance of the CNN in each scenario, which will then be compared to find the best accuracy in line with the objectives of this study.

3. RESULTS AND DISCUSSIONS

The research was conducted according to the steps previously described, and results were obtained for each scenario

TABLE II. Testing results on the CNN scenario with the CNN-1 architecture

Model	P (%)	R (%)	F (%)	Acc (%)
Model 1	72	75	74	73
Model 2	67	83	74	71
Model 3	73	83	78	76
Model 4	69	80	74	72
Model 5	64	9	16	52
Model 6	0	0	0	50
Model 7	49	90	64	48
Model 8	50	89	64	50
Model 9	50	100	67	50
Model 10	0	0	0	50

TABLE III. Testing results on the CNN scenario with the CNN-2 architecture

Model	P (%)	R (%)	F (%)	Acc (%)
Model 1	50	100	67	50
Model 2	0	0	0	50
Model 3	50	100	67	50
Model 4	50	100	67	50
Model 5	80	3	5	51
Model 6	50	100	67	50
Model 7	50	100	67	50
Model 8	0	0	0	50
Model 9	0	0	0	50
Model 10	71	23	34	56

using two different architectures, with each architecture consisting of 10 different models, bringing the total to 20 models. Specifically, for the CNN+LBP+HOG direct input scenario, only one architecture was used, resulting in 10 models.

A. CNN Scenario

The results obtained in the CNN scenario are shown in Table II for the CNN-1 architecture and Table III for the CNN-2 architecture. The best model in this scenario is the third model in the CNN-1 architecture. This model was trained for 60 epochs. After the training process, the accuracy and loss graphs for this model were obtained, as shown in Figure 9 and Figure 10. These graphs indicate that the model did not experience overfitting, as evidenced by the loss value, which tended to decrease during both training and validation, and by the accuracy graphs for both training and validation, which tended to increase. The evaluation results are obtained after testing on 300 images, with an accuracy of 76%, precision of 73%, recall of 83%, and an F1-Score of 78%.

B. CNN + LBP Scenario

The results for the CNN + LBP scenario are shown in Table IV for the CNN-1 architecture and Table V for the CNN-2 architecture. The best model in this scenario is found in the CNN-1 architecture, specifically the third model. This model was trained for 60 epochs, and after the training process, the accuracy and loss graphs were obtained as shown in Figure 11 and Figure 12. These graphs indicate that the model did not experience overfitting or underfitting, as the accuracy graphs for both training and validation tended to increase, and the loss value tended to decrease, although there was no significant decrease before the 15th epoch. The evaluation results are obtained after testing on 300 images, with an accuracy of 78%, precision of 75%, recall of 83%, and an F1-Score of 79%.

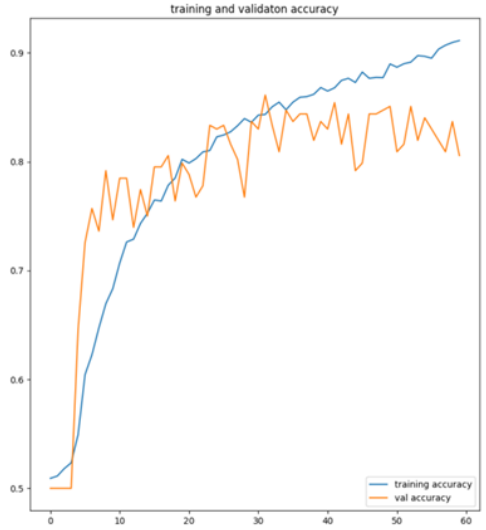

Figure 9. The accuracy graph of the best model in the CNN scenario

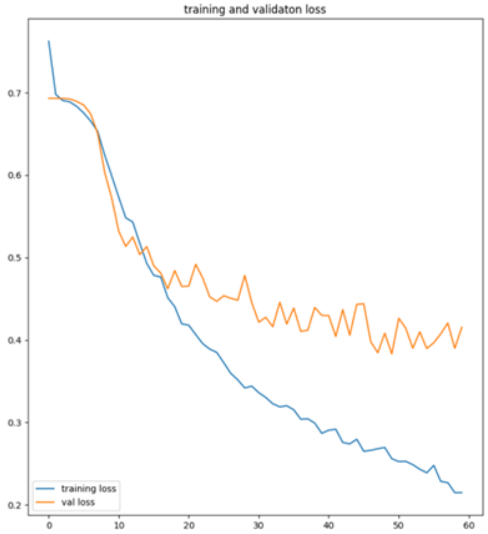

Figure 10. The loss graph of the best model in the CNN scenario

C. CNN + HOG Scenario

The results for the CNN + HOG scenario are shown in Table VI for the CNN-1 architecture and Table VII for the CNN-2 architecture. The best model in this scenario is the third model in the CNN-2 architecture. Training was

TABLE IV. Testing results on the CNN + LBP scenario with the CNN-1 architecture

Model	P (%)	R (%)	F (%)	Acc (%)
Model 1	73	83	78	76
Model 2	74	81	77	76
Model 3	75	83	79	78
Model 4	59	31	40	55
Model 5	51	65	57	52
Model 6	58	45	50	56
Model 7	57	9	15	51
Model 8	55	24	33	52
Model 9	59	28	38	54
Model 10	74	76	75	74

TABLE V. Testing results on the CNN + LBP scenario with the CNN-2 architecture

Model	P (%)	R (%)	F (%)	Acc (%)
Model 1	50	100	67	50
Model 2	50	100	67	50
Model 3	0	0	0	50
Model 4	50	100	67	50
Model 5	56	33	42	54
Model 6	50	100	67	50
Model 7	47	49	48	47
Model 8	50	59	54	50
Model 9	52	8	14	50
Model 10	50	100	67	50

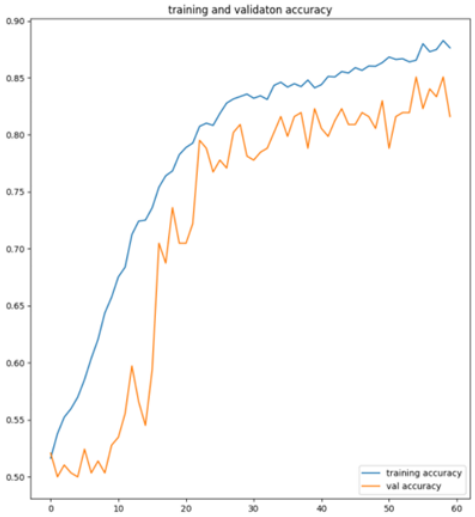

Figure 11. The accuracy graph of the best model in the CNN + LBP scenario

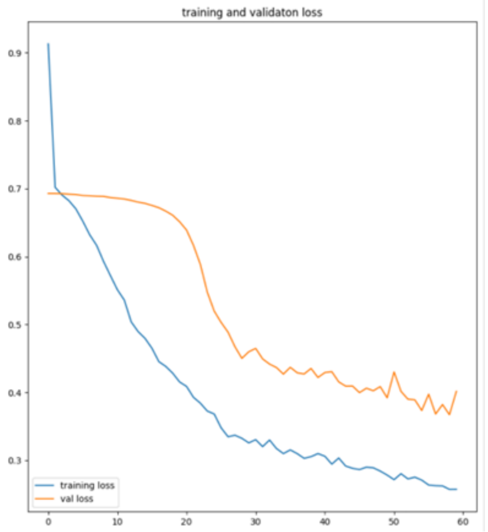

Figure 12. The loss graph of the best model in the CNN + LBP scenario

TABLE VI. Testing results on the CNN + HOG scenario with the CNN-1 architecture

Model	P (%)	R (%)	F (%)	Acc (%)
Model 1	66	81	73	70
Model 2	68	82	75	72
Model 3	70	76	73	71
Model 4	68	79	73	71
Model 5	72	76	74	73
Model 6	74	63	68	71
Model 7	74	66	70	71
Model 8	70	68	69	69
Model 9	74	69	72	73
Model 10	77	65	71	73

conducted for 60 epochs, resulting in accuracy and loss graphs as shown in Figure 13 and Figure 14. These graphs indicate that the model did not experience overfitting or underfitting, as shown by the accuracy graph, which tended to increase, although there was a slight drop and some stagnation in the validation accuracy. However, the loss graph tended to decrease, indicating that the model was able to learn patterns in the available data. The evaluation results are obtained after testing on 300 images, with an accuracy of 75%, precision of 75%, recall of 78%, and an F1-Score of 75%.

D. CNN + LBP + HOG Scenario

The results for the CNN + LBP + HOG scenario are shown in Table VIII for the CNN-1 architecture and Table IX for the CNN-2 architecture. The best model in this scenario is the fourth model in the CNN-2 architecture. This model was trained for 60 epochs, and the resulting accuracy and loss graphs are shown in Figure 15 and Figure 16. These graphs indicate that the model did not experience overfit-

TABLE VII. Testing results on the CNN + HOG scenario with the CNN-2 architecture

Model	P (%)	R (%)	F (%)	Acc (%)
Model 1	70	73	71	71
Model 2	70	73	71	71
Model 3	73	78	75	75
Model 4	75	71	73	73
Model 5	74	67	70	72
Model 6	74	69	71	72
Model 7	73	71	72	72
Model 8	70	67	68	69
Model 9	75	71	73	74
Model 10	73	74	74	73

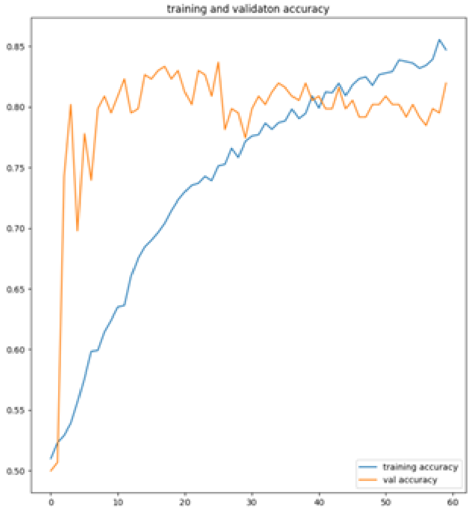

Figure 13. The accuracy graph of the best model in the CNN + HOG scenario

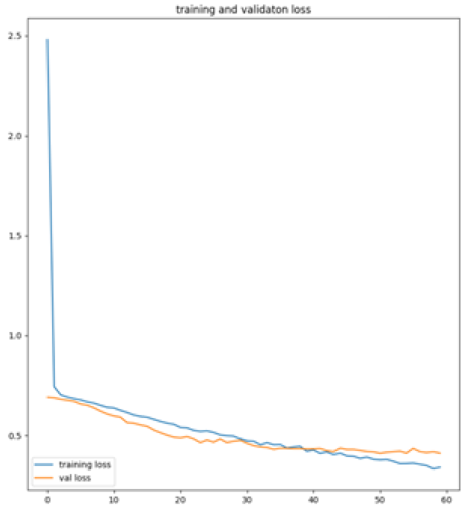

Figure 14. The loss graph of the best model in the CNN + HOG scenario

TABLE VIII. Testing results on the CNN + LBP + HOG scenario with the CNN-1 architecture

Model	P (%)	R (%)	F (%)	Acc (%)
Model 1	73	77	75	74
Model 2	74	78	76	75
Model 3	73	78	75	75
Model 4	73	73	73	73
Model 5	74	80	77	76
Model 6	74	77	75	75
Model 7	74	73	73	74
Model 8	74	67	71	72
Model 9	79	70	74	76
Model 10	74	71	72	73

TABLE IX. Testing results on the CNN + LBP + HOG scenario with the CNN-2 architecture

Model	P (%)	R (%)	F (%)	Acc (%)
Model 1	73	82	77	76
Model 2	74	75	74	74
Model 3	73	82	77	76
Model 4	79	81	80	80
Model 5	76	60	67	70
Model 6	70	85	77	74
Model 7	75	63	68	71
Model 8	77	65	71	73
Model 9	75	63	68	71
Model 10	76	65	70	72

ting or underfitting. It can be observed that the validation accuracy graph fluctuates significantly with each epoch, but the validation loss graph tends to decrease with each epoch, indicating that the model is continuously learning the data patterns. The evaluation results are obtained after testing on 300 images, with an accuracy of 80%, precision of 79%, recall of 81%, and an F1-Score of 80%.

E. CNN + LBP + HOG Direct Input Scenario

The results for the CNN + LBP + HOG Direct Input scenario are shown in Table X for the CNN + LBP + HOG Direct Input architecture, with the best model in this scenario being the fifth model. This model was trained for 36 epochs, and the resulting accuracy and loss graphs are shown in Figure 17 and Figure 18. The graphs indicate that the model was close to overfitting, so early stopping was applied to prevent it. It can be observed that the validation accuracy graph tends to increase, although it occasionally drops, while the validation loss graph generally decreases but rises again in the last epoch, indicating that the model could potentially overfit if training were continued. The evaluation results obtained after testing on 300 images, with an accuracy of 68%, precision of 68%, recall of 69%, and an F1-Score of 68%.

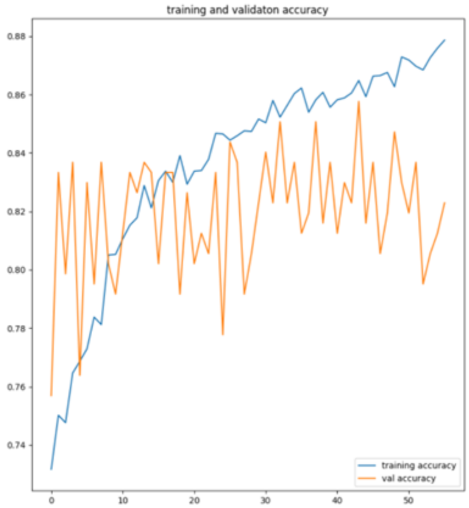

Figure 15. The accuracy graph of the best model in the CNN + LBP + HOG scenario

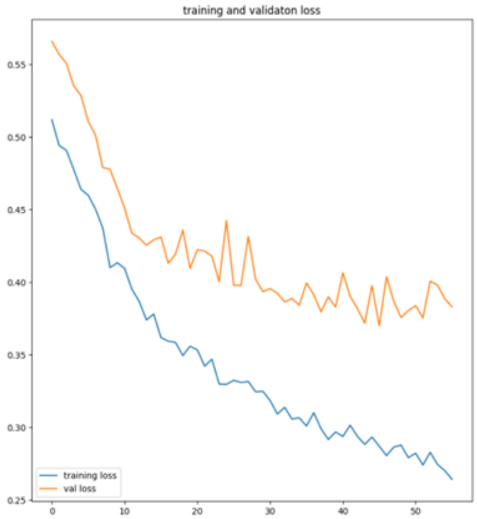

Figure 16. The accuracy graph of the best model in the CNN + LBP + HOG scenario

TABLE X. Testing results on the CNN + LBP + HOG Direct Input scenario

Model	P (%)	R (%)	F (%)	Acc (%)
Model 1	63	63	63	63
Model 2	62	64	63	63
Model 3	65	63	64	64
Model 4	66	66	66	66
Model 5	68	69	68	68
Model 6	71	62	66	68
Model 7	65	57	60	63
Model 8	61	52	56	59
Model 9	62	59	60	61
Model 10	66	57	61	64

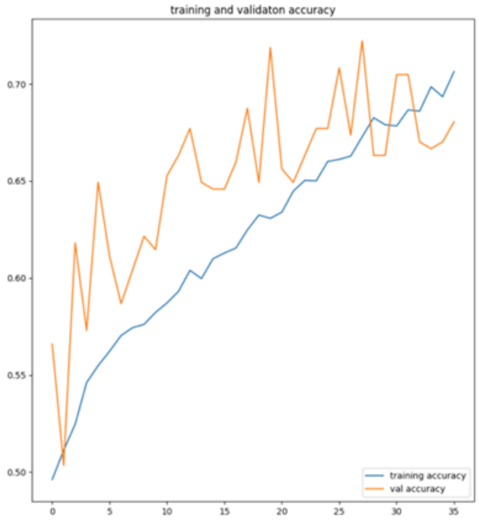

Figure 17. The accuracy graph of the best model in the CNN + LBP + HOG Direct Input scenario

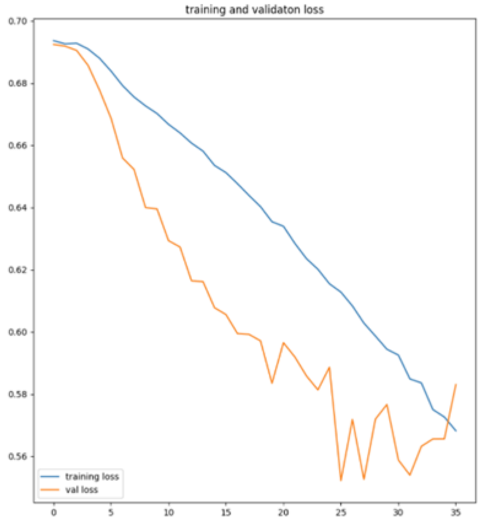

Figure 18. The loss graph of the best model in the CNN + LBP + HOG Direct Input scenario

F. Analysis

Based on the five scenarios that have been conducted, the best results were obtained in the CNN + LBP + HOG scenario with an accuracy of 80%. Compared to the CNN scenario, the LBP feature extraction, which serves as preprocessing, and the combination of LBP and HOG feature fusion are better than not using feature extraction. This aligns with the statement by Simon and Uma (2020), which asserts that if the initial data to be extracted by CNN has already undergone prior extraction, it will facilitate CNN in learning the information, thereby improving CNN's accuracy. The lower accuracy in the CNN + HOG scenario compared to the CNN scenario indicates that feature fusion with images that have not had their features extracted before entering the convolutional layer has not been effective in

TABLE XI. Comparison of accuracy for each scenario

Scenario	Precision(%)	Recall(%)	F1-Score(%)	Accuracy(%)
CNN	73	83	78	76
CNN + LBP	75	83	79	78
CNN + HOG	73	78	75	75
CNN + LBP + HOG	79	81	80	80
CNN + LBP + HOG Direct Input	68	69	68	68

helping CNN identify patterns in faces that can distinguish individuals with autism from those without.

The CNN + LBP + HOG scenario also produced better accuracy compared to the CNN + LBP + HOG Direct Input scenario, even though feature extraction was applied in the CNN + LBP + HOG Direct Input scenario as suggested by Simon and Uma (2020). This indicates that using HOG as a feature extraction method was less effective in assisting the feature extraction process in the CNN convolutional layer, leading to CNN's inability to effectively learn patterns in the faces of individuals with autism. The CNN + HOG scenario also resulted in better accuracy compared to the CNN + LBP + HOG Direct Input scenario, indicating that feature fusion is more appropriately used on facial image datasets for autism detection.

This study concludes that applying LBP and HOG feature extraction to digital image datasets can help improve CNN's accuracy in early autism detection. In this context, LBP serves as preprocessing to assist the CNN convolutional layer in the feature extraction process. The HOG feature extraction, which extracts features that have already been preprocessed by LBP, is used in combination with the features extracted by CNN, thereby enriching the information in the image, which positively impacts CNN's classification accuracy in detecting autism.

4. CONCLUSION AND SUGGESTION

A. *Conclusion*

This study found that the combined feature extraction using Local Binary Pattern (LBP) and Histogram of Oriented Gradient (HOG) can improve the accuracy of the CNN model in the case of autism detection using a digital facial image dataset. This demonstrates that the use of combined feature extraction can aid the learning process of the model by providing important information present in facial images, which positively impacts the model's accuracy. Additionally, the combination of feature extraction methods resulted in better accuracy compared to using only one feature extraction method in this case. However, this combination must be applied correctly to avoid making it difficult for the CNN to recognize patterns for classification, as this could lead to a decrease in accuracy.

B. *Suggestion*

Future research could build upon the findings of this study by exploring more complex and advanced model architectures, as well as larger and more diverse datasets. Enhancing the diversity of the dataset in terms of quantity, racial, and ethnic representation would likely improve the generalizability of the model. Additionally, future studies could investigate other combinations of feature extraction techniques to further improve the accuracy and robustness of the model. Experimenting with different architectures and methodologies could lead to the development of more effective models for the task at hand.

References

[1] F. R. Volkmar, R. Paul, S. J. Rogers, and K. A. Pelphrey, "Handbook of autism and pervasive developmental disorders, fourth edition," Mar. 2014. [Online]. Available: http://dx.doi.org/10.1002/9781118911389

[2] M. Maulana and I. Muhsin, *Anak Autis: Mendidik Anak Autis dan Gangguan Mental Lain Menuju Anak Cerdas dan Sehat*, cetakan ii, 2017, cetakan 7, 2014 ed. Yogyakarta: Kata Hati, 2017, text in Indonesian.

[3] 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, no. 1, p. 15, 2011. [Online]. Available: http://dx.doi.org/10.1186/2040-2392-2-15

[4] G. Tripi, S. Roux, D. Matranga, L. Maniscalco, P. Glorioso, F. Bonnet-Brilhault, and M. Roccella, "Cranio-facial characteristics in children with autism spectrum disorders (asd)," *Journal of Clinical Medicine*, vol. 8, no. 5, p. 641, May 2019. [Online]. Available: http://dx.doi.org/10.3390/jcm8050641

[5] A. Subasi, *Feature Extraction and Dimension Reduction*. Elsevier, 2019, p. 193–275. [Online]. Available: http://dx.doi.org/10.1016/b978-0-12-817444-9.00004-0

[6] O. Modi and A. Subasi, *Breast tumor detection in ultrasound images using artificial intelligence*. Elsevier, 2023, p. 137–181. [Online]. Available: http://dx.doi.org/10.1016/B978-0-443-18450-5.00003-7

[7] S. Z. Jumani, F. Ali, S. Guriro, I. A. Kandhro, A. Khan, and A. Zaidi, "Facial expression recognition with histogram of oriented gradients using cnn," *Indian Journal of Science and Technology*, vol. 12, no. 24, p. 1–8, Jun. 2019. [Online]. Available: http://dx.doi.org/10.17485/ijst/2019/v12i24/145093

[8] M. F. Rabbi, S. M. M. Hasan, A. I. Champa, and M. A. Zaman, "A convolutional neural network model for early-stage detection of autism spectrum disorder," in *2021 International Conference on Information and Communication Technology for Sustainable Development (ICICT4SD)*. IEEE, Feb. 2021.

[9] S. S. Jaffar and H. A. Abdulbaqi, "Facial expression recognition in

static images for autism children using cnn approaches," in *2022 Fifth College of Science International Conference of Recent Trends in Information Technology (CSCTIT)*. IEEE, Nov. 2022.

[10] M. S. Alam, M. M. Rashid, R. Roy, A. R. Faizabadi, K. D. Gupta, and M. M. Ahsan, "Empirical study of autism spectrum disorder diagnosis using facial images by improved transfer learning approach," *Bioengineering*, vol. 9, no. 11, p. 710, Nov. 2022. [Online]. Available: http://dx.doi.org/10.3390/bioengineering9110710

[11] S. Ganesan, R. Dr., and D. S. J, "Prediction of autism spectrum disorder by facial recognition using machine learning," *Webology*, vol. 18, no. 02, p. 406–417, Sep. 2021. [Online]. Available: http://dx.doi.org/10.14704/WEB/V18SI02/WEB18291

[12] Y. Khosla, P. Ramachandra, and N. Chaitra, "Detection of autistic individuals using facial images and deep learning," in *2021 IEEE International Conference on Computation System and Information Technology for Sustainable Solutions (CSITSS)*. IEEE, Dec. 2021.

[13] L. A. Rahman and P. Marikannan Booma, "The early detection of autism within children through facial recognition; a deep transfer learning approach," in *2022 2nd International Conference on New Technologies of Information and Communication (NTIC)*. IEEE, Dec. 2022.

[14] Q. U. Ain, H. Al-Sahaf, B. Xue, and M. Zhang, "Automatically diagnosing skin cancers from multimodality images using two-stage genetic programming," *IEEE Transactions on Cybernetics*, vol. 53, no. 5, p. 2727–2740, May 2023. [Online]. Available: http://dx.doi.org/10.1109/TCYB.2022.3182474

[15] P. Pankaja Lakshmi., M. Sivagami., and V. Balaji., "A novel lt-lbp based prediction model for covid-ct images with machine learning," in *2021 International Conference on Information Systems and Advanced Technologies (ICISAT)*. IEEE, Dec. 2021.

[16] I. M. R. Albakri, M. I. Ahmad, M. N. Md Isa, and M. Z. N. Al-Dabagh, "Face recognition based on fusion feature extraction algorithms," in *2024 International Conference on Emerging Smart Computing and Informatics (ESCI)*. IEEE, Mar. 2024.

[17] P. Simon and U. V, "Deep learning based feature extraction for texture classification," *Procedia Computer Science*, vol. 171, p. 1680–1687, 2020. [Online]. Available: http://dx.doi.org/10.1016/j.procs.2020.04.180

[18] H. O. Salau, O. A. Abisoye, I. O. Oyefolahan, and S. A. Adepoju, "Enhanced chest x-ray classification model for covid-19 patients using hog and lbp," in *2022 5th Information Technology for Education and Development (ITED)*. IEEE, Nov. 2022.

[19] https://www.kaggle.com/cihan063/autism-image-data.

[20] T. Akter, M. H. Ali, M. I. Khan, M. S. Satu, M. J. Uddin, S. A. Alyami, S. Ali, A. 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, p. 734, May 2021. [Online]. Available: http://dx.doi.org/10.3390/brainsci11060734

[21] K. K. Mujeeb Rahman and M. M. Subashini, "Identification of autism in children using static facial features and deep neural networks," *Brain Sciences*, vol. 12, no. 1, p. 94, Jan. 2022. [Online]. Available: http://dx.doi.org/10.3390/brainsci12010094