Mobile-Based Serious Game for Facial Expression Recognition Training in Children with Autism Spectrum Disorder Using MobileNet Model

Sarah Hanifah Pontoh
Information Technology Department
Institut Teknologi Sepuluh Nopember
Surabaya, Indonesia
5027201006@student.its.ac.id

Rizka Wakhidatus Sholikah
Information Technology Department
Institut Teknologi Sepuluh Nopember
Surabaya, Indonesia
wakhidatus@its.ac.id

Hatma Suryotrisongko
Information Technology Department
Institut Teknologi Sepuluh Nopember
Surabaya, Indonesia
hatma@is.its.ac.id

Abstract—**The human ability to understand someone's emotions through facial expressions plays a crucial role in everyday social interactions. However, in children with autism spectrum disorder (ASD), this ability is often impaired, affecting their social interactions. Studies have shown that children with ASD are well-acquainted with and often engage with technology, including smartphones and video games. This research proposes a solution in the form of a mobile-based serious game application as an educational tool to assist children with autism in learning human emotions through facial expressions. Serious games are considered appropriate educational tools because it leverages four pillars of learning: attention, active learning, feedback, and consolidation. The gamification technique used involves a snake and ladder game that prompts users (children with autism) to mimic the facial expressions displayed in the game. The game also measures the accuracy of the user's facial expressions using MobileNet model, then uses it as scores within the gameplay to enhance user engagement. This model reached the highest validation accuracy of 0.89 with training accuracy of 0.88. Final evaluation of the model is conducted using test set and resulted in an accuracy of 0.83.**

Keywords—ASD, serious game, facial expression recognition, MobileNet

I. Introduction

Humans generally have an intuitive ability to understand someone's facial expressions and feelings. This ability allows us to read other people's emotions and adjust our responses when communicating directly. However, for children with autism, this ability is often impaired or hindered. According to the WHO, approximately 1 in 100 children worldwide has autism spectrum disorder [1]. In Indonesia, the Ministry of Health states that the number of autistic children increases by about 500 children every year, reaching 2.4 million in 2024 [2]. Children with autism tend to have difficulties in recognizing basic facial emotions overall and in recognizing all individual emotions [3]. This can cause them to have difficulties in social interaction, communication, and integrating both with emotions [4].

A solution to help children with autism overcome difficulties in recognizing facial expressions can be realized using technology. Currently, there are many mobile applications developed as therapeutic tools for children with autism. 90% of children with autism are visual learners [5]. Most of them are also familiar with gadgets such as smartphones [6] and enjoy playing video games due to its visual appeal [7]. Therefore, serious games or educational games, can be a suitable solution to help children with ASD to learn facial expressions.

The development of a serious game application as an educational therapy tool for children with autism has been conducted previously in an article titled “A Mobile Game Platform for Improving Social Communication in Children with Autism: A Feasibility Study” [8]. This image guessing game was tested on a group of 72 children with autism for 4 weeks and resulted in an increase of 3.97 points on the Social Responsiveness Scale-2 (SRS-2) and 5.27 points on the Vineland Adaptive Behavior Scales-II (VABS-II) Socialization Standard Score.

Based on the above explanation, this research proposes a more holistic serious game application that can help children with autism to learn someone's emotions from facial expressions.

II. Literature Review

In recent years, numerous studies have focused on developing technology-based educational tools for children with ASD. One study [9] developed a facial expression recognition system in the form of a mobile web application to help train children with autism to interpret someone's emotions using a deep learning model to assess the accuracy of displayed facial expressions. This application performed well in recognizing facial expressions, achieving highest accuracy score of 0.91. However, this web application has not yet incorporated visually appealing gamification methods that could increase children's interest in its continued use.

In addition to research [8], another research [10] proposed a serious game called SimpleTEA, which engages nonverbal children with ASD to learn basic everyday vocabulary in several categories, including food, emotions, numbers, and daily activities, through guessing game. Although this research has not been directly tested on children with autism, the proposed application received positive feedback from therapists for its potential to support everyday communication for nonverbal children with autism.

Serious games can serve as an effective educational tool by utilizing four pillars of learning: attention, active learning, feedback, and consolidation. Attention in learning is necessary to create enthusiasm. This can be achieved by using attractive game visuals, sounds, or music. Active learning encourages users to continuously learn through interactive activities in the game, for example, by creating a storyline. Feedback is needed as a response to what the user does in the game, such as scoring or rewards when tasks are performed correctly. Consolidation helps users to remember what they have learned previously in the long term, for example, by applying task repetition in the game [11].

To implement these learning pillars, this research proposes a mobile-based serious game that uses CNN model to measure user facial expression accuracy and use this accuracy score as the game score to enhance user engagement. CNN models are commonly used for facial expression recognition (FER) tasks, such as in research [9] which utilized VGG16. In 2021, another study [12] also developed a mobile application for facial expression recognition using MobileNet model. The model used in this research achieved an accuracy of 85%.

III. Methodology

A. *Dataset*

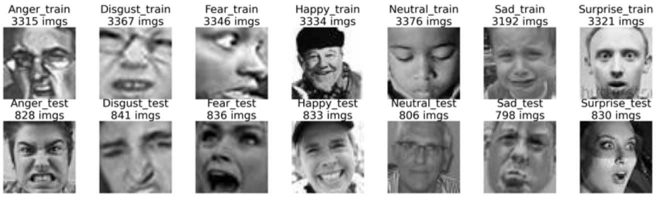


Figure 1. FER2013

The FER2013 (Facial Expression Recognition 2013) dataset is one of the most widely used datasets for human facial expression classification tasks. The dataset used in this research is an augmented version of FER2013 by Ilia Prizker. This version has a more balanced data distribution across each class compared to the original FER2013 dataset. The training set contains a total of 23,251 images, while the test set consists of 5,772 images, each with a resolution of 48 x 48 pixels. Each set includes images representing seven types of facial expressions: angry, disgust, fear, happy, neutral, sad, and surprised [13].

B. *Facial Expression Recognition Model*

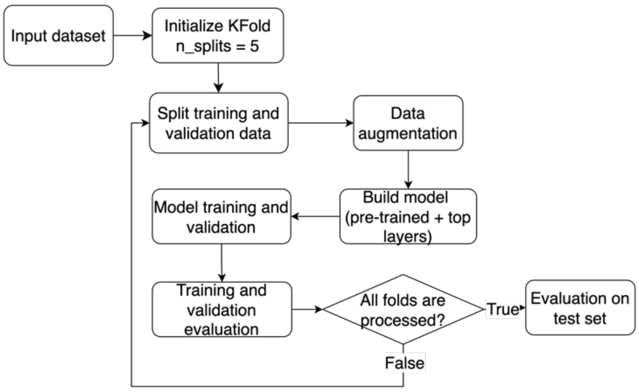

Input dataset
Initialize KFold n_splits = 5
Split training and validation data
Data augmentation
Build model (pre-trained + top layers)
Model training and validation
Training and validation evaluation
All folds are processed?
True
False
Evaluation on test set

Figure 2. Modelling process

The model creation process began by inputting the FER2013 dataset. Then, by using k-fold cross-validation, the entire modeling process was performed on each fold. K-fold cross-validation was used to ensure that the model learned from the entire training data evenly, rather than focusing on static training and validation portions as typically done in traditional training and validation split.

K-fold cross-validation divided the training set into *n* parts (folds), with each 1/*n* part taking turns as the validation subset (orange blocks in Figure 3) while the remaining parts were used as training subsets. This research used 5 folds to train and validate the model.

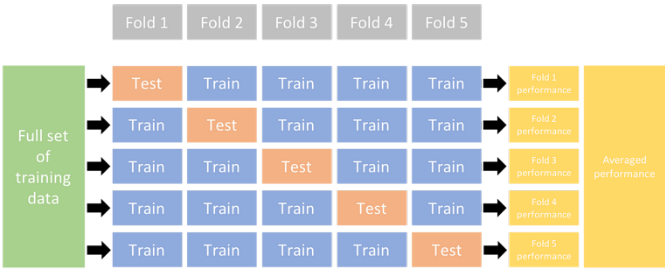


Figure 3. K-fold cross validation [18]

In each fold, data was split and then augmented to increase the diversity of the training subset, allowing the model to learn more information from each image. The image augmentation applied in this process includes:

1. **Rescale**: Normalized pixel values from a range of 0-255 to 0-1. This was done to make the data smaller and easier to process.
2. **Rotation_range**: The rotation was kept to a minimum to maintain upright orientation of facial images. Excessive distortion in the data could make it more difficult for the model to recognize facial features.
3. **Width_shift_range** and **height_shift_range**: Shifted the image vertically and horizontally.
4. **Shear_range**: Shifted part of the image while keeping the other intact. An example of shear is transforming a square shape into a parallelogram, as if some corners are being pulled vertically or horizontally.
5. **Horizontal_flip** and **vertical_flip**: Vertical flip was not done to keep the upright orientation.
6. **Fill_mode**: Filled new empty pixels when the image was shifted or rotated during previous augmentations. 'nearest' was used to match the new pixels with the nearest pixel values.

Model was then generated by using a pre-trained CNN model as its base, which was responsible as feature extractor to learn characteristics from each image. This research compared two types of pre-trained models, VGG16 and MobileNet, to determine which model performs the best.

1) *VGG16*

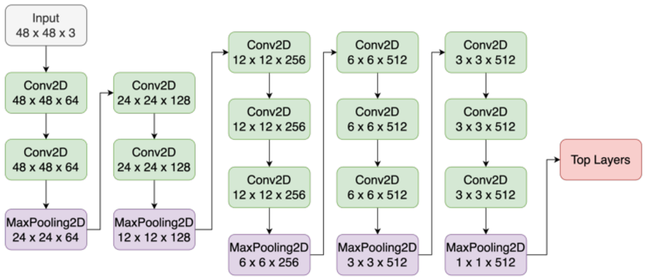


Figure 4. VGG16 used in this research

Visual Geometry Group 16 (VGG16) is one of CNN models from Keras that utilizes 16 pre-trained layers on the ImageNet dataset, consisting of 14 million images categorized into 1000 classes. The architecture of this model includes 13 convolutional layers, 5 pooling layers, and 3 fully connected (FC) layers, with the last FC layer having 1000 channels for classifying image classes. This model is widely used for tasks in image recognition due to its ability to recognize various types of images [14].

The input size used for this model is $48 \times 48 \times 3$ to match the dataset. The filters size used in the convolutional layers is 3×3, with a stride of 1, keeping the spatial dimensions of the output the same (48×48) after passing through one convolutional layer. The number of filters in convolutional block I is 64, resulting in an output shape of $48 \times 48 \times 64$. Block I end with a pooling layer with a 2×2 window and a stride of 2, reducing the spatial dimensions of the input to half its original size $(24 \times 24 \times 64)$. This convolution and pooling architecture are repeated, with the number of filters doubling with each subsequent block, reaching 512 filters in block 5, resulting in a feature map with 512 channels. The top layers consist of fully connected layers, including several dense layers.

2) MobileNet

This model belongs to the family of efficient models characterized by minimal parameter usage while still achieving maximum model performance. This approach reduces computational load, making the model suitable for devices with limited resources such as smartphones and embedded systems. To reduce computational cost by minimizing parameters, MobileNet uses depthwise separable convolutions, which divide the convolution process into two parts: depthwise convolution and pointwise convolution.

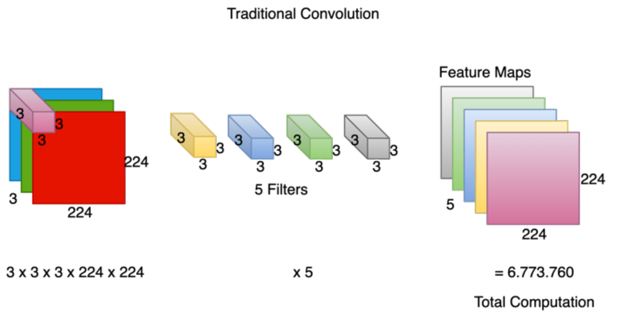


Figure 5. Traditional convolution

In Figure 5, if an input $D_F \times D_F \times M$ where D_F is the height and width and M is the number of channels (depth) of the input, uses filters with a kernel size of $D_K \times D_K \times M \times N$ where D_K is the height and width of the filter and N is the number of filters, the resulting feature map will have a size of $D_F \times D_F \times N$ (assuming stride = 1 and padding = 1). With this traditional process, total computation can be calculated as:

$$D_K \cdot D_K \cdot M \cdot D_F \cdot D_F \cdot N \tag{1}$$

The total parameters of this traditional convolutional layer can be calculated based on the filter (kernel) size as follows:

$$D_K \times D_K \times M \times N \tag{2}$$

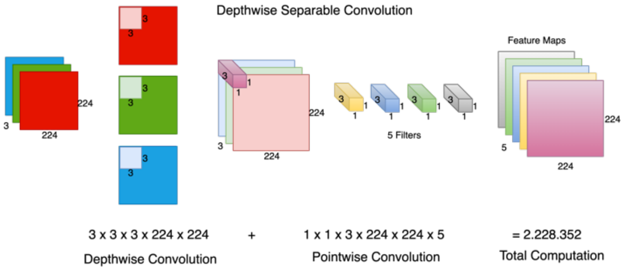


Figure 6. Depthwise separable convolution

The difference between traditional convolution and depthwise separable convolution lies in the computation process used to produce the same feature map. In the depthwise convolution layer, the convolution process is performed separately by $D_K \times D_K \times 1$ filter on each input channel. The pointwise convolution layer then combines and processes the output from the depthwise convolution using $1 \times 1 \times M$ filters, with N filters in total, resulting in a feature map of $D_F \times D_F \times N$ (assuming stride = 1 and padding = 1). Therefore, when calculated, the total computation using depthwise separable convolution is:

$$D_K \cdot D_K \cdot M \ \cdot D_F \cdot D_F \ + M \cdot D_F \cdot D_F \cdot N \tag{3}$$

With the total parameters of:

$$D_K \times D_K \times M + M \times N \tag{4}$$

This computational cost is more lightweight compared to traditional convolution because it reduces multiplication operations, making the model more efficient [15], as shown in Figure 5 and Figure 6.

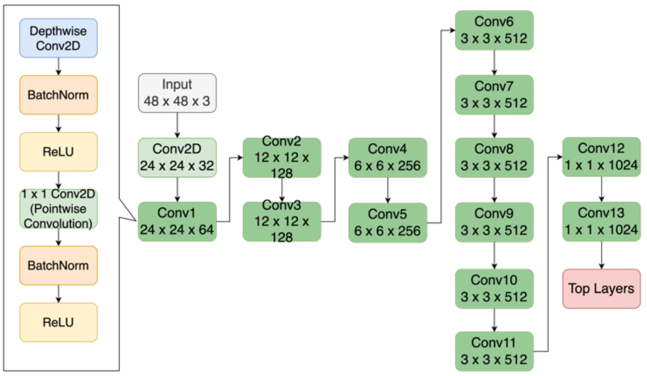


Figure 7. MobileNet model used in this research

The pre-trained MobileNet accepts an input of $48 \times 48 \times 3$ (modified to match dataset) and starts with a single traditional convolutional layer with 32 filters and stride = 2, resulting in an output shape that is half the size of the input, $24 \times 24 \times 32$. In the next 13 convolutional layer blocks, each block consists of a depthwise convolutional layer with a 3×3 filter, followed by a batch normalization layer and the first ReLU activation, then a pointwise convolutional layer with 1×1 filters, with N filters in total, and a second batch normalization layer and ReLU activation. In some blocks, down sampling is performed to reduce the spatial dimensions of the input from the previous block by using stride = 2 in the depthwise convolutional layer, such as in block 2 after receiving input from block 1. The number of filters in the pointwise convolutional layer determines the number of channels in the output feature map, as in block 1, where the pointwise layer uses 64 filters, resulting in an output shape of $24 \times 24 \times 64$. The number of filters in the model increases with each block so that the model can learn more information from the input. Each block uses batch normalization layers to prevent overfitting, maintain stability, and speed up the training process [16], and the ReLU activation function to add non-linearity to the network so it can learn complex pattern from the image [17].

Top layers for classification are appended to the base model. The whole model then undergoes training and validation and is evaluated based on the metrics of accuracy, precision, recall, and F1 Score.

Table 1. Top layers on both models

Layer	**Output Shape**
flatten	(None, 1024)
batch_normalization	(None, 1024)
dense	(None, 1024)
batch_normalization_1	(None, 1024)
activation_1 ReLU	(None, 1024)
dropout_1 0.5	(None, 1024)
dense_1	(None, 512)
batch_normalization_2	(None, 512)
activation_2 ReLU	(None, 512)
dropout_2 0.5	(None, 512)
dense_2	(None, 128)
batch_normalization_3	(None, 128)
activation_3 ReLU	(None, 128)
dropout_3 0.5	(None, 128)
dense_3	(None, 32)
batch_normalization_4	(None, 32)
activation_4 ReLU	(None, 32)
dense_4	(None, 7)

C. Serious Game Development

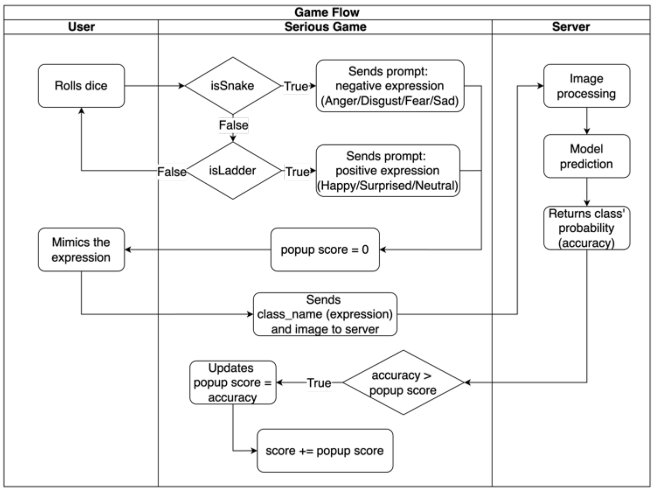

Game Flow
User
Serious Game
Server
Rolls dice
isSnake
True
Sends prompt: negative expression (Anger/Disgust/Fear/Sad)
False
isLadder
False
True
Sends prompt: positive expression (Happy/Surprised/Neutral)
Image processing
Model prediction
Returns class' probability (accuracy)
Mimics the expression
popup score = 0
Sends class_name (expression) and image to server
Updates popup score = accuracy
True
accuracy > popup score
score += popup score

Figure 8. Game flow

The proposed serious game was designed in the form of a snakes and ladders game that prompted user to mimic expressions displayed in the application through pop-up prompts. These prompts appear as pop-up screens containing the name and illustration of the facial expression the user needs to mimic. The expressions required by the pop-up are chosen randomly according to the user's True condition. Pop-up starts a video to record the user mimicking the facial expression. During this process, video frames are continuously captured with a 2-second delay. The application then sends the user's facial images and the name of the mimicked expression to the model in server via API. Before it is fed into the model, image processing is done on the server, then model predicts the facial expression shown by the user and measures the accuracy (probability) of the requested facial expression class. This accuracy score is used as the score or "reward" in the game. With each pop-up occurrence, this accuracy score is continuously updated (every 2 seconds) on each image prediction request made by the model. The final accuracy score used as the result of one pop-up occurrence is the highest value from all the requests sent by that pop-up prompt. After the pop-up ends, the player's score is updated by adding the previous score with the current pop-up score. This gamification method is used on this research because of its ease of use, and to increase user engagement with the game, considering the target users are children with autism.

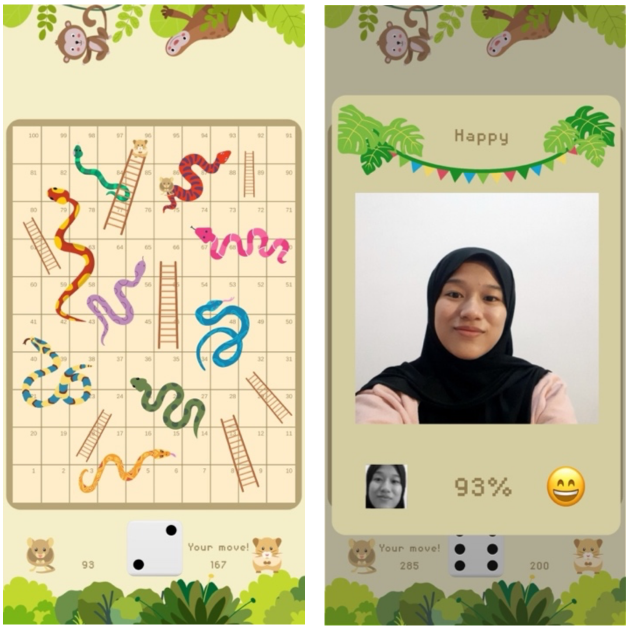


Figure 9. Game interface

D. Model and App Integration

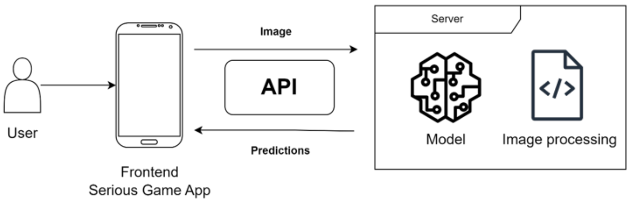


Figure 10. System architecture

From the diagram above (Figure 10), the CNN model was deployed on a server. This approach was done to facilitate future model maintenance, such as updates and bug fixes. Additionally, this centralized model ensures consistent performance, unaffected by the smartphone capabilities of individual users. The application communicates with the model and sends prediction requests via an API. Similarly, the model sends responses back in the form of prediction results, including the accuracy of the user's facial expression input.

E. Testing

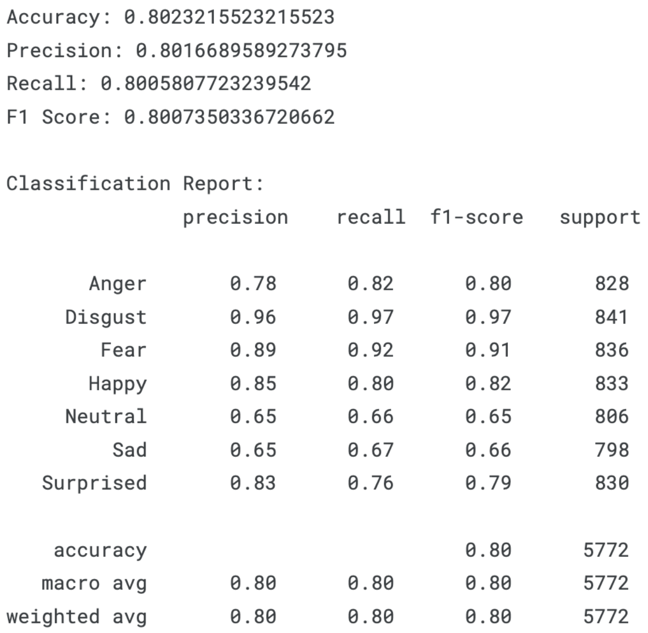
Testing was done to assess two main aspects of the application: the CNN model used for FER and the performance of the serious game when used by the users. The CNN model was tested using 8 test scenarios, and usability testing for the app was conducted with children with ASD to gather feedback.

IV. Results and Discussion

A. *Model Evaluation*

Table 2. VGG16 training and validation result

Fold	***Epoch***	***Training Accuracy***	***Validation Accuracy***
1	5	0,14	0,14
	10	0,14	0,14
2	5	0,14	0,14
	10	0,15	0,14
	15	0,2	0,19
	20	0,23	0,19
	25	0,28	0,24
	30	0,34	0,35
3	5	0,39	0,42
	10	0,47	0,47
	15	0,55	0,56
	20	0,6	0,6
	25	0,65	0,64
	30	0,68	0,68
4	5	0,7	0,7
	10	0,74	0,71
	15	0,77	0,74
	20	0,79	0,75
	25	0,82	0,76
	30	0,84	0,77
5	5	0,8	0,79
	10	0,82	0,79
	15	0,85	0,82
	20	0,87	0,82
	25	0,9	0,83



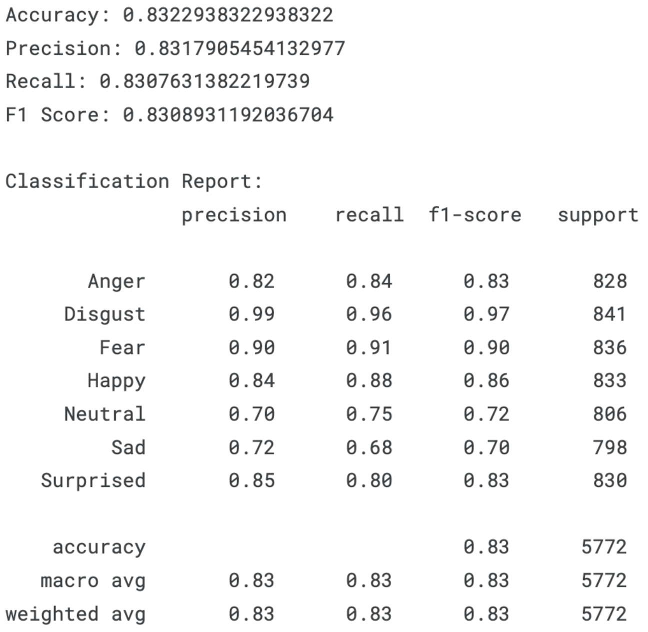
```
Accuracy: 0.8023215523215523
Precision: 0.8016689589273795
Recall: 0.8005807723239542
F1 Score: 0.8007350336720662

Classification Report:
              precision    recall  f1-score   support

       Anger       0.78      0.82      0.80       828
     Disgust       0.96      0.97      0.97       841
        Fear       0.89      0.92      0.91       836
       Happy       0.85      0.80      0.82       833
     Neutral       0.65      0.66      0.65       806
         Sad       0.65      0.67      0.66       798
   Surprised       0.83      0.76      0.79       830

    accuracy                           0.80      5772
   macro avg       0.80      0.80      0.80      5772
weighted avg       0.80      0.80      0.80      5772
```

Figure 11. VGG16 on test set

Both models underwent training and validation with 23,251 data points using the k-fold method. During the training process of the model with VGG16, the accuracy value increased with the rise in epochs, indicating effective learning by the model. The training and validation accuracy of the model showed no significant gap, implying minimal overfitting. The VGG16 model achieved the highest validation accuracy of 0.837 with a training accuracy of 0.90 at fold 5, epoch 27. During the final evaluation with a test set of 5,772 data points, the VGG16 model achieved similar accuracy, precision, recall, and F1 scores of 0.8, demonstrating consistent predictions.

Table 3. MobileNet training and validaiton result

Fold	***Epoch***	***Training Accuracy***	***Validation Accuracy***
1	5	0,54	0,55
	10	0,64	0,67
	15	0,69	0,75
	20	0,73	0,74
	25	0,75	0,77
	30	0,77	0,76
2	5	0,78	0,79
	10	0,8	0,78
	15	0,81	0,8
	20	0,82	0,81
	25	0,85	0,81
	30	0,86	0,82
3	5	0,83	0,82
	10	0,84	0,84
	15	0,87	0,86
	20	0,88	0,86
	25	0,89	0,86
	30	0,9	0,86
4	5	0,85	0,85
	10	0,87	0,88
5	5	0,85	0,88
	10	0,88	0,88
	15	0,89	0,88
	20	0,9	0,89



```
Accuracy: 0.8322938322938322
Precision: 0.8317905454132977
Recall: 0.8307631382219739
F1 Score: 0.8308931192036704

Classification Report:
              precision    recall  f1-score   support

       Anger       0.82      0.84      0.83       828
     Disgust       0.99      0.96      0.97       841
        Fear       0.90      0.91      0.90       836
       Happy       0.84      0.88      0.86       833
     Neutral       0.70      0.75      0.72       806
         Sad       0.72      0.68      0.70       798
   Surprised       0.85      0.80      0.83       830

    accuracy                           0.83      5772
   macro avg       0.83      0.83      0.83      5772
weighted avg       0.83      0.83      0.83      5772
```

Figure 12. MobileNet on test set

Meanwhile, the MobileNet model performed better than the previous model. The accuracy of MobileNet model in the first fold was significantly higher than the VGG16 model, at 0.55, and continued to improve with the increase in epochs. MobileNet achieved the highest validation accuracy of 0.89 with a training accuracy of 0.88 at fold 5, epoch 12. Similar to VGG16 model, the final evaluation of MobileNet model also showed consistent prediction results but with higher values, with accuracy, precision, recall, and F1 scores of 0.83. Based on these results, we decided to use MobileNet model in the system.

The app with MobileNet model was then tested in 8 scenarios with various environmental conditions to assess the model's performance in identifying facial expressions. The variables used as test conditions were background (plain and noisy), the use of glasses, and the distance between the face and the camera (± 20 cm and > 30 cm). Among the 8 scenarios, the most ideal environmental condition to obtain maximum accuracy on each expression was scenario 4, which involved a plain background, no glasses, and a face-to-camera distance of more than 30 cm.

Table 4. Scenario 4

Scenario 4: Plain background, no glasses, distance > 30 cm						
Anger	Disgust	Fear	Happy	Neutral	Sad	Surprised
94%	97%	77%	100%	91%	98%	100%

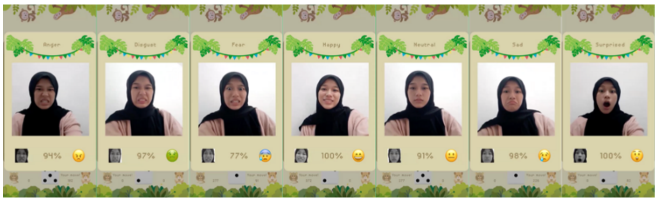


Figure 13. Ideal condition

A plain background reduces noise in the image, allowing the model to make more accurate predictions. The use of glasses also significantly affects the model's predictions, as they can obscure facial features like eyebrows, which are important for predicting classes like anger and disgust. Distance also influences the model's prediction results. The model tends to provide higher prediction probability when the face is more than 30 cm away from the camera. This is because the dataset used to train the model consists of images with a size of 48 x 48, making the model more accustomed to small-sized and low-resolution facial images. If the face-to-camera distance is too close, the face cascade will capture images with a resolution higher than the data the model was trained on. From the accuracy results of the seven classes, the model can predict the happy, sad, and surprised classes very accurately and quite well in predicting the anger and neutral classes. However, for the disgust and fear classes, the model's accuracy is still low in some scenarios.

Table 5. Average accuracy from scenario 1-8

Anger	Disgust	Fear	Happy	Neutral	Sad	Surprised
73,9	61,0	59,5	99,8	79,5	91,3	99,6

B. *Serious Game Evaluation*

Usability testing was conducted with 3 respondents with ASD. Respondent X was a nonverbal autistic child who did not yet understand facial expressions, while respondents Y and Z had already learned facial expressions at their school.

In terms of engagement with the game, nonverbal respondent was more often distracted and needed more assistance to follow the game's flow compared to the verbal respondents. This can be seen from a shorter playing duration of the nonverbal respondent compared to the verbal respondents. However, it is also important to note that this duration was influenced by the number of pop-up occurrence during the game. In terms of learnability, respondent X still needed assistance in mimicking expressions, while respondents Y and Z understood the game flow better and could mimic facial expressions to the best of their ability.

Table 6. Usability testing

Respondents	Expression							
	A	*D*	*F*	*H*	*N*	*Sad*	*Sp*	Duration
X (13 y/o, nonverbal)		0 %		8%	4%			12' 15"
Y (17 y/o, verbal)	12%	0 %	0 %	100%	93%	1%	0%	20' 12"
Z (10 y/o, nonverbal)	91%				9%	66%	0%	14' 9"

V. Conclusion

This research developed a mobile-based serious game application to help children with autism learn and practice facial expressions. The developed serious game takes the form of a two-player snakes and ladders game that prompts the user to mimic expressions displayed in the application through pop-up prompts. This game uses a CNN model with MobileNet as its base model to measure the accuracy of the facial expressions mimicked by the user. There are 7 types of expressions used in the game and in training the model: angry, disgust, fear, happy, neutral, sad, and surprised. The model achieved the highest validation accuracy of 0.89 with a training accuracy of 0.88. The model's evaluation on test set showed an accuracy of 0.83. The game and the model interact via API to communicate in real-time.

Scenario based testing was conducted to determine the most ideal condition to achieve the highest score in the game. Highest accuracy was achieved in an environment with a plain background, no glasses, and a face-to-camera distance of more than 30 cm. From the test results, the model could predict all expressions well, but the probability for class disgust and fear remained relatively low. The app was also tested with children with autism, and the feedback from parents and teachers was positive, stating that the application is effective in helping autistic children learn facial expressions.

References

[1] World Health Organization, "Autism," World Health Organization. Accessed: Jul. 01, 2024. [Online]. Available: https://www.who.int/news-room/fact-sheets/detail/autism-spectrum-disorders

[2] D. Stefanni, "Wamenkes Ungkap 2,4 Juta Anak di Indonesia Idap Autisme," detikHealth. Accessed: Jul. 01, 2024. [Online]. Available: https://health.detik.com/berita-detikhealth/d-7336606/wamenkes-ungkap-2-4-juta-anak-di-indonesia-idap-autisme

[3] M. K. Yeung, "A systematic review and meta-analysis of facial emotion recognition in autism spectrum disorder: The specificity of deficits and the role of task characteristics," 2022. doi: 10.1016/j.neubiorev.2021.104518.

[4] C. Papoutsi, A. Drigas, and C. Skianis, "Mobile applications to improve emotional intelligence in Autism - A review," 2018, *International Association*

of Online Engineering. doi: 10.3991/ijim.v12i6.9073.

[5] L. A. Hodgdon, "Visual Strategies for Improving Visual Communication: Volume I: Practical support for school and home. ," *Quirk Roberts Publishing*, 1999.

[6] Z. S. De Urturi, A. M. Zorrilla, and B. G. Zapirain, "Serious Game based on first aid education for individuals with Autism Spectrum Disorder (ASD) using android mobile devices," in *Proceedings of CGAMES'2011 USA - 16th International Conference on Computer Games: AI, Animation, Mobile, Interactive Multimedia, Educational and Serious Games*, 2011. doi: 10.1109/CGAMES.2011.6000343.

[7] O. B. Hansen, A. Abdurihim, and S. McCallum, "Emotion recognition for mobile devices with a potential use in serious games for autism spectrum disorder," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2013, pp. 1–14. doi: 10.1007/978-3-642-40790-1_1.

[8] Y. Penev *et al.*, "A Mobile Game Platform for Improving Social Communication in Children with Autism: A Feasibility Study," *Appl Clin Inform*, vol. 12, no. 5, 2021, doi: 10.1055/s-0041-1736626.

[9] S. L. C. Nugroho, R. W. Sholikah, and R. V. H. Ginardi, "Pengembangan Aplikasi Mobile untuk Pengenalan Emosi Manusia Menggunakan Facial Expression Recognition untuk Membantu Anak dengan Autisme," *Jurnal Teknik ITS*, 2023, Accessed: Mar. 10, 2024. [Online]. Available: http://repository.its.ac.id/id/eprint/96628

[10] A. Jaramillo-Alcázar, J. Arias, I. Albornoz, A. Alvarado, and S. Luján-Mora, "Method for the Development of Accessible Mobile Serious Games for Children with Autism Spectrum Disorder," *Int J Environ Res Public Health*, vol. 19, no. 7, Apr. 2022, doi: 10.3390/ijerph19073844.

[11] D. Drummond, A. Hadchouel, and A. Tesnière, "Serious games for health: three steps forwards," Dec. 01, 2017, *BioMed Central Ltd*. doi: 10.1186/s41077-017-0036-3.

[12] N. A. S. Badrulhisham and N. N. A. Mangshor, "Emotion Recognition Using Convolutional Neural Network (CNN)," in *Journal of Physics: Conference Series*, 2021. doi: 10.1088/1742-6596/1962/1/012040.

[13] Y. Khaireddin and Z. Chen, "Facial Emotion Recognition: State of the Art Performance on FER2013," Boston, MA, USA, May 2021.

[14] Z. Song, K. Nguyen, T. Nguyen, C. Cho, and J. Gao, "Spartan Face Mask Detection and Facial Recognition System," *Healthcare (Switzerland)*, vol. 10, no. 1, 2022, doi: 10.3390/healthcare10010087.

[15] A. G. Howard *et al.*, "MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications," Apr. 2017, [Online]. Available: http://arxiv.org/abs/1704.04861

[16] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *32nd International Conference on Machine Learning, ICML 2015*, 2015.

[17] V. Nair and G. E. Hinton, "Rectified linear units improve Restricted Boltzmann machines," in *ICML 2010 - Proceedings, 27th International Conference on Machine Learning*, 2010.

[18] B. Boehmke and B. Greenwell, *Hands-On Machine Learning with R*. 2019. doi: 10.1201/9780367816377.