Frontiers in Health Informatics
ISSN-Online: 2676-7104
www.healthinformaticsjournal.com
2024; Vol 13: Issue 6
Open Access

AI-based Emotional Learning System for Individuals with Autism Spectrum Disorder

Hoyeol Yang [1], KyoungOck Park [2], Jeong Tak Ryu [3], Yoosoo Oh [4] , Kyuman Jeong [5]

[1,5] School of AI, Daegu University

[2] Department of Elementary Special Education, Daegu University

[3] Department of Electronic and Electrical Engineering, Daegu University

[4] School of Computer & Information Engineering, Daegu University

Cite this paper as: Hoyeol Yang , Kyoung Ock Park , Jeong Tak Ryu , Yoosoo Oh , Kyuman Jeong (2024) AI-based Emotional Learning System for Individuals with Autism Spectrum Disorder. Frontiers in Health *Informatics*, *13(6) 1015-1025*

Abstract. Autism spectrum disorder (ASD) is a neurodevelopmental condition characterized by social communication difficulties and restricted, repetitive patterns of behavior. Individuals with ASD often experience challenges with emotion regulation, which can lead to a variety of negative consequences, including anxiety, depression, and social isolation. Artificial intelligence (AI) has the potential to be a valuable tool for developing interventions to address emotion regulation difficulties in ASD. AI-based emotion regulation tools could provide individuals with ASD with personalized feedback and support, helping them to identify, understand, and manage their emotions. In this paper, we discuss the development of an AI-based emotion regulation tool for ASD. The tool utilizes a combination of machine learning and natural language processing techniques to identify and classify emotions from facial expressions and text-based communication. The tool also provides personalized feedback and support to users, helping them to develop effective emotion regulation strategies. We present a preliminary evaluation of the tool, which demonstrates its ability to accurately identify emotions from facial expressions and text-based communication. We also discuss the potential benefits of the tool for individuals with ASD.

Keywords: Artificial Intelligence, Emotional Learning System, , Autism Spectrum Disorder

1. Introduction

Autism Spectrum Disorder (ASD) is a neurodevelopmental disorder characterized by limitations in social communication and interaction and repetitive [1].

People with autism spectrum disorder have difficulty understanding the emotions of others because they lack the ability to read emotions from their facial expressions [2].

The ability to recognize emotions from facial expressions plays a very important role in social interaction and interpersonal relationships. Children who have difficulty developing and utilizing this ability effectively may experience various challenges and isolation [3,4].

Although existing treatment methods in society may be effective, they are still time-consuming and expensive, and many children may not receive appropriate treatment due to the limitations of treatment institutions and accessibility issues.Since there are problems that can occur later in life, such as mental problems and difficulties in social adaptation if proper treatment or support is not provided during the growth period, we aimed to develop an easy-to-use emotional learning system based on facial expression image recognition to solve these problems.

Since the basic acquisition method and knowledge are different depending on the home environment and living conditions in which they lived, the system was developed with an easy approach, and the advantage of this emotional learning system is that anyone can easily use the system to improve interactions and communication with their guardians in daily life, and this can also help children with autism spectrum disorder, their families, teachers, and those who need other treatment and support. It is cheaper than existing treatment methods, and more people can use it through a smart app. This can have the effect of reducing limited treatment institutions and financial burdens, and the emotional learning system based on facial expression image recognition provides users with autism spectrum disorder with the opportunity to learn and develop on their own. It is important to find out the ability and method of how to process vocabulary while looking at this language characteristic from the semantic aspect of high-functioning autistic children, but it is also important to look at how vocabulary is acquired and the pattern can be broadened, so that we can gain a basis for understanding the overall characteristics, which is the reason we decided to do this research.

In this paper, we proposed an emotional learning application based on facial expression image recognition [5-14] that is made into a playful learning device to help children with high-functioning autism spectrum disorders, which is less expensive than existing treatment methods and can be utilized by more people through the use of a smart app. This can have the effect of reducing the limited treatment institutions and financial burden, and the emotional learning system based on facial expression image recognition can provide users with autism spectrum disorders with opportunities to learn and develop on their own, which can form relationships between parents and their high-functioning autistic children, and allow parents to relax while using the application.

2. Related Work

The biggest difficulty caused by autism spectrum disorder (hereinafter referred to as 'autism') is the problem of communicating with others. In order to foster this communication ability, which is the most basic for forming sociality, the warm application (hereinafter referred to as 'app') 'Look At Me' was created. Look at Me is an app that 'trains' children to communicate with others. Through the Look at Me app, children with autism complete six missions and one bonus mission. The time required for this is 20 minutes per day. Each mission becomes more difficult as it progresses. If your child finds it too difficult, you can skip the mission. However, as your child's communication skills improve as they pass more difficult missions, it is best to avoid skipping too much. One of the most important things when communicating with others is understanding their emotions. If you are laughing alone when the other person is angry or sad, communication will inevitably be cut off. Children with autism have difficulty because they have a low ability to empathize with the emotions of others.

Children with autism spectrum disorder often need help from their guardians, and much research and development is being conducted to solve the biggest problem, which is communication with others. The Look at Me app developed by Samsung is a representative app that trains people how to communicate with others.

'One Step at a Time' is a training exercise where you look at four randomly arranged pictures and match the order of the facial expressions. The user has to correct the order in which a blank face gradually changes into a face expressing a specific emotion. The assignment will be given in four rounds. Two of them will be solved using photos of your own expressions, and the remaining two will be solved using photos saved in the app. Through this task, users not only learn how to make facial expressions of others, but also learn how to make their own facial expressions naturally. Through the process of understanding their own facial expressions, they can better understand the facial expressions of others and the emotions contained in them.

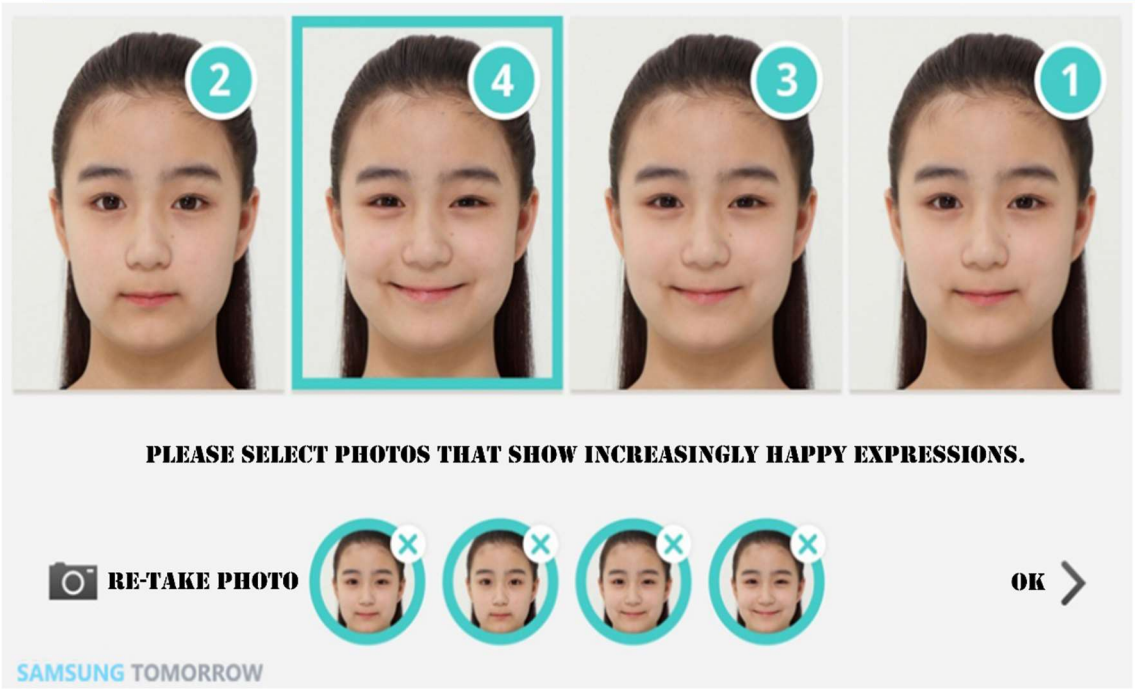


Fig. 1: Expression Reading Training_'One Step at a Time'

Another difficulty that autistic children face is the inability to properly express their emotions. Since they are unable to express their emotions accurately, others have no way of knowing how they feel, and as a result, a wall is created between them and they often suffer losses in their group life.

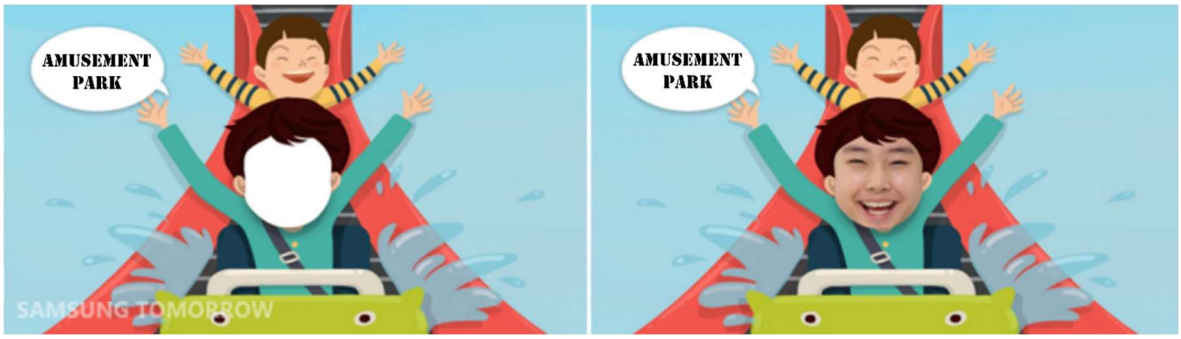


Fig. 2: Training to express emotions and intentions_'What expression should I use in this situation?'

'What expression should I use in this situation?' is a training task that helps users express their emotions. When the program starts, a blank face appears on the screen. Various situations are provided, such as joy, sadness, anger, disappointment, and laughter. The child must match their face to the blank face and take a picture. If this training is repeated, the user will gradually be able to express their intentions appropriately for the situation.
When living in a group, there are times when you need to understand the atmosphere. If you don't, you can easily end up being treated as an outsider. 'Identifying the Mood' is a training task developed based on this very fact. When the task begins, nine faces appear on the left and right sides of the screen for a short time. Each face has a different expression. The user must guess which side has more people making a certain facial expression

(happy, scared, angry, etc.) after the screen disappears. This training develops the ability to intuitively read the emotions of many people and naturally mingle with the crowd.

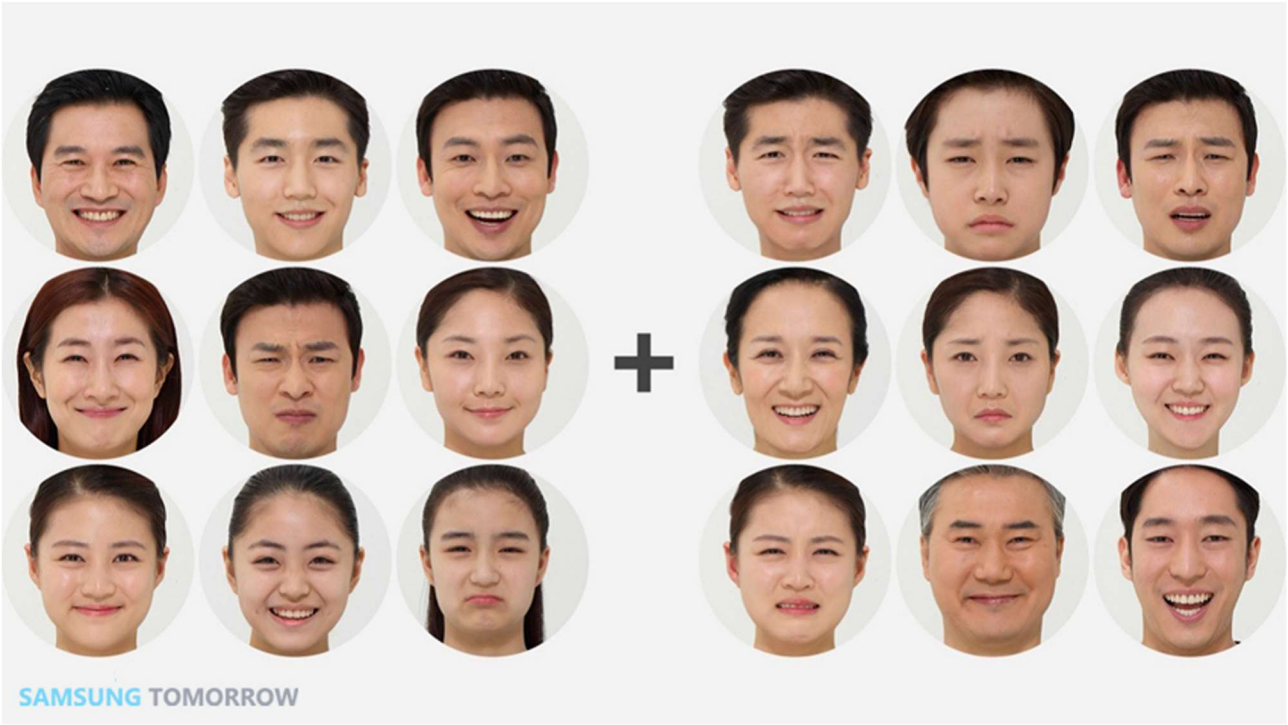


Fig. 3: Training to understand other people's emotions 'Understanding the atmosphere'

3. Proposed Algorithm

3.1. Building training data

The learning data used in this project was “Composite Images for Korean Emotion Recognition” in AIHUB. [https://zrr.kr/1SbU] The learning data is image data that stores facial expressions for emotion analysis by identifying the contextual information of the scene. The images are classified into a total of 7 categories: joy, embarrassment, anger, anxiety, hurt, sadness, and neutral, and consist of source data and label data as shown in Table 1. The source data consists of 500,000 cases, including about 250,000 data from professional actors who are good at expressing emotions and about 250,000 data from ordinary people. Joy, embarrassment, anger, anxiety, hurt, sadness, and neutral emotions are evenly distributed, and about 70,000 images are included for each category. Five emotions were used as learning data by integrating sadness and hurt and excluding neutrality for emotion recognition.

Table 1: Emotion Category

Category	Detailed Emotion
anger	Grumbling, frustrated, irritated, defensive, malicious, impatient, disgusting, angry, annoying
sorrow	Disappointed, heartbroken, regretful, depressed, numb, cynical, tearful,

	defeated, disillusioned
unrest	Fearful, stressed, vulnerable, confused, embarrassed, skeptical, worried, cautious, nervous
wound	Jealous, betrayed, isolated, shocked, needy, victimized, wronged, distressed, abandoned
embarrassment	Isolated, self-conscious, lonely, inferior, guilty, shameful, disgusting, pathetic, confused
delight	grateful, believing, comfortable, satisfied, excited, relaxed, relieved, excited, confident
neutrality	expressionless, emotionless

For the learning data, 18,000 out of 20,000 images were used for each category. Since the images include both places and people, faces were recognized using the pre-trained Haar cascades, a feature-based object detection algorithm provided by OpenCV, in order to recognize only human expressions, and saved in a file. Images that failed to be recognized among the separately saved face images were deleted. After that, the image suitable for the VGG16 model was 224X224 in size, but it was resized to 96X96 due to computer memory issues. After adjusting the images, the learning data was separated into a learning dataset, a validation dataset, and a test dataset, and consisted of 68,812 training data, 14,742 validation data, and 13,921 test data. Image augmentation does not always prevent overfitting in model learning, but since we are learning images with new data, I thought it would be good to add more diverse images, so I used ImageGenerator to process ImageAugmentation on the images. The preprocessed images do not have labels, but flow_from_directory was used to attach labels to the images according to the file name.

```
def data_augmentation(self, img_path, emotion):
    def random_noise(x):
        x = x + np.random.normal(size=x.shape) * np.random.uniform(1, 5)
        x = x - x.min()
        x = x / x.max()

        return x * 255.0

    datagen = ImageDataGenerator(
        rotation_range=40,
        width_shift_range=0.07,
        height_shift_range=0.07,
        shear_range=0.2,
        zoom_range=0.2,
        horizontal_flip=True,
        fill_mode='nearest',
        preprocessing_function=random_noise
    )
```

Fig. 4: Pseudo code of input data processing

In the experiment, image augmentation was performed using rescale, rotation_range, width_shift_range, height_shift_range, shear_range, zoom_range, horizontal_flip, fill_mode, and validation_split as shown in Fig 5.

Fig. 5: Result of image augmentation

3.2. Learning Model

The learning model used in this task is VGG16. VGG stands for Visual Geometry Group, and 16 refers to the number of layers in the model as shown in Fig 6. VGG16 consists of a total of 16 layers, of which 14 are

convolutional layers and 3 are fully connected layers.

ConvNet Configuration					
A	A-LRN	B	C	D	E
11 weight layers	11 weight layers	13 weight layers	16 weight layers	16 weight layers	19 weight layers
input (224 × 224 RGB image)					
conv3-64	conv3-64 **LRN**	conv3-64 **conv3-64**	conv3-64 conv3-64	conv3-64 conv3-64	conv3-64 conv3-64
maxpool					
conv3-128	conv3-128	conv3-128 **conv3-128**	conv3-128 conv3-128	conv3-128 conv3-128	conv3-128 conv3-128
maxpool					
conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256 **conv1-256**	conv3-256 conv3-256 **conv3-256**	conv3-256 conv3-256 conv3-256 **conv3-256**
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 conv3-512 **conv3-512**
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 conv3-512 **conv3-512**
maxpool					
FC-4096					
FC-4096					
FC-1000					
soft-max					

Fig. 6: Configuration of ConvNet

This model has a deep but simple structure, and improves performance by replacing the large kernel-size filter used in AlexNet with multiple 3X3 kernel-size filters. And it achieved 92.7% top-5 test accuracy on ImageNet, a dataset consisting of more than 14 million images belonging to 1,000 classes. Despite its simple structure, the VGG16 model shows excellent performance in various computer vision tasks, and is also suitable for transfer learning. The VGG16 model, pre-trained on a large-scale dataset, can be used for various computer vision tasks, and shows high performance even on small datasets.

We chose VGG16 to proceed with the project, believing that using the pre-trained VGG16 model on ImageNet, a large-scale dataset, would yield high accuracy despite being a new dataset.

```
input_tensor = Input(shape=(96, 96, 3), dtype='float32', name='input')
pre_trained_vgg = VGG16(weights='imagenet', include_top=False, input_shape=(96, 96, 3))

for layer in pre_trained_vgg.layers[:13]:
    layer.trainable = False
for layer in pre_trained_vgg.layers[13:]:
    layer.trainable = True
```

Fig. 7: VGG16 code

VGG16 is trained on the premise of a color image of 224X224 size, but due to memory limitations, the input was processed in the form of (96, 96, 3). When the size becomes 96X96, the detail decreases, which may show a relatively low accuracy phenomenon. After loading the model so that the weights of 'imagenet' can be used by using the model trained with 'imagenet', layers before the 13th layer were frozen and layers after the 13th layer were unfrozen to perform fine-tuning. Since fine-tuning does not define the number of layers, the layer that showed good performance when checked by reducing it one by one was shown at the 13th layer.

```
from keras import models, layers, regularizers, optimizers
from tensorflow.keras.initializers import HeNormal

additional_model = models.Sequential()
additional_model.add(pre_trained_vgg)
additional_model.add(layers.Flatten())

additional_model.add(layers.Dense(256, kernel_regularizer = regularizers.l2(l2=0.001),activity_regularizer=regularizers.l1(l1=0.001), activation='relu',  kernel_initializer=HeNormal()))
additional_model.add(layers.BatchNormalization())
additional_model.add(layers.Dropout(0.4))

additional_model.add(layers.Dense(128, kernel_regularizer = regularizers.l2(l2=0.001),activity_regularizer=regularizers.l1(l1=0.001), activation='relu',  kernel_initializer=HeNormal()))
additional_model.add(layers.BatchNormalization())
additional_model.add(layers.Dropout(0.4))

additional_model.add(layers.Dense(64, kernel_regularizer = regularizers.l2(l2=0.001),activity_regularizer=regularizers.l1(l1=0.001), activation='relu',  kernel_initializer=HeNormal()))
additional_model.add(layers.BatchNormalization())
additional_model.add(layers.Dropout(0.4))

additional_model.add(layers.Dense(5, activation='softmax'))

additional_model.compile(loss='sparse_categorical_crossentropy',
                         optimizer=optimizers.Adam(learning_rate=0.0001),
                         metrics=['acc'])
```

Fig. 8: Transfer learning code

The above Fig 8 is mainly used in transfer learning, which is performed to create a new model based on a specific layer in an existing model and to improve the performance of the model by adding multiple layers. This was used because it can show good performance when applying a previously learned model to new data. The final model was compiled with the optimizer set to Adam and the learning rate set to 0.0001.

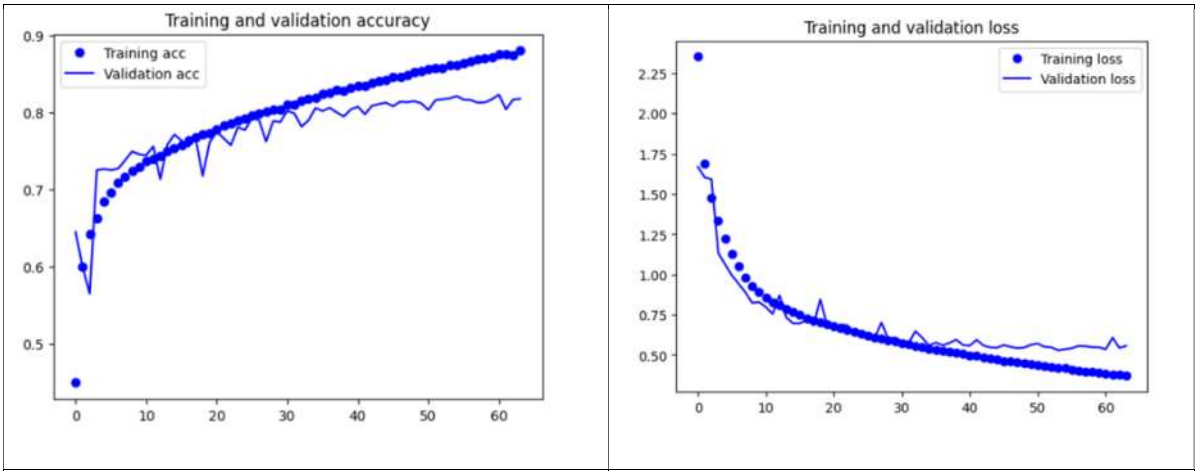
```
model_checkpoint = ModelCheckpoint(filepath='thirdtest_model.keras', monitor='val_loss', save_best_only=True)
early_stopping = EarlyStopping(monitor='val_loss', patience=10)

callbacks = [early_stopping, model_checkpoint]


history = additional_model.fit(
    train_data_gen, validation_data=val_data_gen,
    epochs=100, batch_size=32, callbacks=callbacks)
```

Fig. 9: Epoch and batch size for the learning porcess

Learning was performed for 100 epochs with the batch size set to 32 as shown in Fig 9. Early termination was set so that learning could stop before overfitting occurred as shown in Fig 10.



Accuracy of training data and validation data	Loss of training data and validation data

Fig. 10: Result of training process

4. Result

4.1. Experimental Setup

First, when you select an emotion to learn from the five emotions, in the 'Create Expression' step, you create an image of the selected emotion and the correct expression by combining the eyebrows, eyes, and mouth. If you create an expression that does not match the emotion, you try again, and if successful, you move on to the next step, 'Imitate Expression'. In the 'Imitate Expression' step, you use a camera to imitate the expression you learned previously. The captured image is analyzed using the learning model, and if the expression of the learned emotion and the expression in the captured image match, the correct answer screen is displayed. In addition, if you select a different image or make an expression that does not match, you can take a picture again.

4.2. Performance Evaluation

When the final saved learning model was input into the new data, the test data, it showed an accuracy of 81.37%, and the resulting confusion matrix was as follows.

Fig. 11: Accuracy result

```
from tensorflow.keras.models import load_model

model = load_model('/content/thirdtest_model.keras')

loss, accuracy = model.evaluate(test_generator)
print(f'Test accuracy: {accuracy * 100:.2f}%')

218/218 [==============================] - 13s 59ms/step - loss: 0.5450 - acc: 0.8137
Test accuracy: 81.37%
```

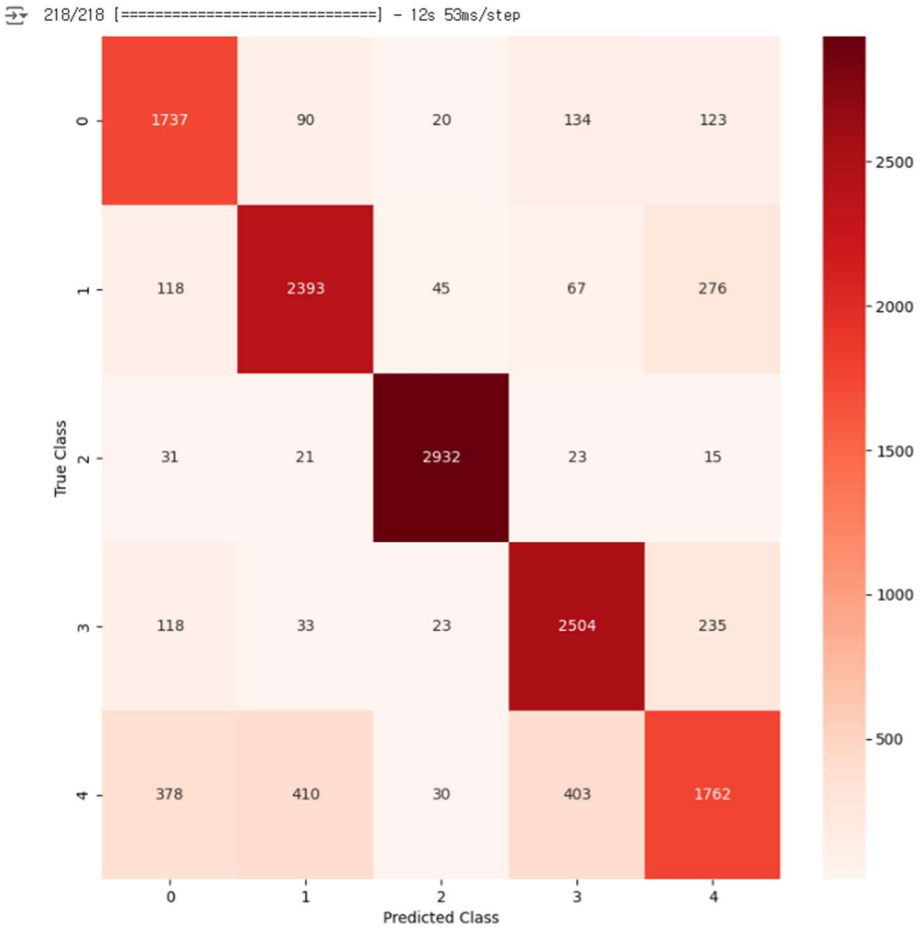


Fig. 12: Confusion matrix

5. Conclusion

The facial expression image recognition-based emotional learning system can be easily used by anyone, eliminating the inconvenience caused by limited treatment institutions and financial burden. This provides more opportunities for those who need treatment and support, and emphasizes the importance of early care. In particular, children with autism spectrum disorder often need the help of their guardians, so we developed an app that allows autistic children to learn on their own, allowing their guardians to rest for even just 5 minutes while they learn. When autistic children learn, their guardians can check how many times they tried and which expressions they learned the most, allowing them to see which expressions their children are interested in, and helping them to provide appropriate guided learning. In addition, since it targets children with high-functioning autism spectrum disorder, it increases accessibility for children with familiar shapes and an easy-to-access UI, providing opportunities for them to learn and develop on their own without the help of their guardians, and providing personalized learning achievement records after the expression-matching game to help them continue to grow. As a result, the app we developed increases the activity of children with autism spectrum disorder and improves the quality of life for children and their families.

6. Acknowledgements

This work was supported by the Ministry of Education of the Republic of Korea and the National Research Foundation of Korea (NRF-2022S1A5C2A07091326).

7. References

[1] J. Lee, and K Chung, "An Investigation of Complex Emotion Recognition Using Video Clips in Adolescents with and without Autism Spectrum Disorders", Journal of the Korean Association for Persons with Autism, vol. 19, no. 3, pp. 27-50, 2019.

[2] S. H. Kim, M. A. Hwang, and S. H. Ko, "Characteristics of Pronoun Comprehension by Type and Implicit Causality of Interpersonal Verbs in Children With High-Functioning Autism Spectrum Disorder", Journal of Speech-Language & Hearing Disorders, vol. 27, no. 1, pp. 45-54, 2018.

[3] J. Hwang, and J. Choi, "Meta-analysis of Smart Devices Utilization Intervention for Students with Autism Spectrum Disorder", Journal of the Korean Association for Persons with Autism, vol. 20, no. 2, pp. 87-111, 2020.

[4] J. Kang, and Y. Lee, "Smart Devices for People with Autism Spectrum Disorders", Journal of the Korean Association for Persons with Autism, vol. 14, no. 2, pp. 93-118, 2014.

[5] E. Chanang, H. Deshpande, and C. Bergler, "Facial expression space learning", Proceedings of the 10th Pacific Conference on Computer Graphics and Applications, pp. 68-76, 2002.

[6] Y. Chang, C. Hu, R. Feris, and M. Turk, "Manifold based analysis of facial expression", Image and Vision Computing, vol. 24, no. 6, pp. 605-614, 2006.

[7] C. Shan, S. Gong, and P. McOwan, "Appearance manifold of facial expression", IEEE International Workshop on Human-Computer Interaction, LNCS 3766, pp. 221-230, 2005.

[8] I. Cohen, N. Sebe, L. Chen, A. Garg, and T. Huang, "Facial expression recognition from video sequences: temporal and static modeling", Computer Vision and Image Understanding, vol. 91, no. 1, pp. 160-187, 2003.

[9] N. Sebe, M. Lew, Y. Sun, I. Cohen, T. Gevers, and T. Huang, "Authentic facial expression analysis", Image and Vision Computing, vol. 25, no. 12, pp. 1856-1863, 2007.

[10] Y. Jiangsheng, "Method of k-Nearest Neighbors", Institute of Computational Linguistics, Peking University, China, 2002.

[11] G. Edwards, C. Taylor, and T. Cootes, "Active appearance models", IEEE Transaction on Pattern Analysis and Machine Intelligence, vol. 23, no. 5, pp. 681-685, 2001.

[12] I. Matthews and S. Baker, "Active appearance models revisited", International Journal of Computer Vision, vol. 60, no. 2, pp. 135-164, 2004.

[13] A. Elgammal and C. Lee, "Separating style and content on a nonlinear manifold", Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, pp. 520-527, 2004.

[14] J. Tenenbaum, V. Silva, and J. Langford, "A global geometric framework for nonlinear dimensionality reduction", Science, vol. 290, no. 5500, pp. 2319-2323, 2000.