Intelligent Automation & Soft Computing

Tech Science Press

DOI: 10.32604/iasc.2023.030674
Article

Acknowledge of Emotions for Improving Student-Robot Interaction

Hasan Han[1], Oguzcan Karadeniz[1], Tugba Dalyan[2,*], Elena Battini Sonmez[2] and Baykal Sarioglu[1]

[1]Department of Electrical and Electronics Engineering, Istanbul Bilgi University, Istanbul, 34060, Turkey
[2]Department of Computer Engineering, Istanbul Bilgi University, Istanbul, 34060, Turkey
*Corresponding Author: Tugba Dalyan. Email: tugba.yildiz@bilgi.edu.tr
Received: 30 March 2022; Accepted: 25 May 2022

Abstract: Robot companions will soon be part of our everyday life and students in the engineering faculty must be trained to design, build, and interact with them. The two affordable robots presented in this paper have been designed and constructed by two undergraduate students; one artificial agent is based on the Nvidia Jetson Nano development board and the other one on a remote computer system. Moreover, the robots have been refined with an empathetic system, to make them more user-friendly. Since automatic facial expression recognition skills is a necessary pre-processing step for acknowledging emotions, this paper tested different variations of Convolutional Neural Networks (CNN) to detect the six facial expressions plus the neutral face. The state-of-the-art performance of 75.1% on the Facial Expression Recognition (FER) 2013 database has been reached by the ensemble voting method. The runner-up model is the Visual Geometry Group (VGG) 16 which has been adopted by the two robots to recognize the expressions of the human partner and behave accordingly. An empirical study run among 55 university students confirmed the hypothesis that contact with empathetic artificial agents contributes to increasing the acceptance rate of robots

Keywords: Artificial intelligence; deep learning; convolutional neural networks; facial expression recognition; robotics; education; professional learning

1 Introduction

Robots are ubiquitous in our life. Among the main tasks assigned to remote virtual agents are search and rescue [1–3], military applications [4,5], and space exploration [6–8], while proximate robots are designed to be used at home, for distraction, enjoyment, or support [9–12], in offices, for assistance [13,14], in a classroom, to engage students in education [15–18], in a hotel, to act as receptionists [19,20], in hospitals, to support patients and for diagnosis [21–23] and in other public environments, for accompanying people such as visitors in a museum [24–26].

That is, robots are part of our everyday life and proximate robots require the creation of realistic artificial agents, which is a challenging issue currently tackled in the artificial intelligence research area, i.e., several studies aim to make robots easy to use and capable of engaging with humans [27–30]. A common denominator among the mentioned studies is the need to increase the sensitivities of robots, which can be implemented by making virtual agents capable of recognizing emotions and acting accordingly.

A parallel and complementary approach is supported by the hypothesis that frequent and close contact with technology helps people to mature positive attitudes toward virtual agents as well as to develop mental models that influence their interaction. That is, while it is necessary to improve the physical and psychological skills of robots, it is also desirable to train humans to interface with them to trigger their mental models that affect interaction with robots, i.e., the commonly accepted hypothesis that people with "gaming experience" is more likely to interact better with robots, if only for their positive attitude towards technology.

Focusing on education, robotics is an excellent tool for engineering students to develop their practical skills, and boost their creativity and teamwork expertise [31]. This work presents to the research community two affordable robots that have been designed and implemented by undergraduate students at the Electrical and Electronics Department of Istanbul Bilgi University. The robots are refined with an empathetic system which makes them more user-friendly. The qualitative evaluation proved the efficacy of acknowledging emotion in increasing the acceptance rate of artificial agents among university students.

To summarize, the main contributions of this paper are: (1) To present two robots designed and built by students of the Engineering Faculty, without any extra funding; (2) To detail the implementation of a successful FER system; (3) To discuss the efficacy of our attempt in increasing positive attitude toward virtual agents among university students; (4) To compare the pros and cons of edge computing-based and remote computer-based robot systems.

The paper is organized as follows: Section 2 describes the related works on automatic FER systems; Section 3 overviews the main databases of FER with particular attention to FER2013; Section 4 describes several models which have been trained to create an automatic FER system; Section 5 introduces the two robots built ad-hoc; Section 6 details the experimental setup and discusses the results; Section 7 illustrates the results of the surveys run among undergraduate students. Finally, Section 8 concludes the paper.

2 State-of-the-Art Performance on FER

The acknowledgment of the emotions expressed by the human partner is a desirable feature of proximate robots, which may contribute to increasing their level of acceptability. The necessary preprocessing step is FER. This section overviews related work on automatic FER; all cited papers are challenging the FER2013 database. The current state-of-the-art has been reached by [32], which proposed a Convolutional Neural Network (CNN) based model that achieved 75.2% accuracy. Moreover, the paper [32] shows algorithmic differences and their effects on performance. Their best results were reached with Stochastic Gradient Descent (SGD) optimizer, momentum (0, 9), batch size (128), learning rate (0, 1), and weight decay (0.0001). The authors of [32] also modified classical models such as VGG, Inception, and Residual Neural Network (ResNet) to reach the accuracy of 72.7%, 71.6%, and 72.4%, respectively. Still, on the FER2013 database, the study [33] presented a model inspired by GoogLeNet and AlexNet architectures. While AlexNet reached the top accuracy of 61.1%, the model proposed in [33] achieved 66.4%. Work [34] presented four different architectures

to address the problem of FER based on the VGG model. Due to the large number of data in the FER2013 dataset, the authors decreased the number of filters of all convolutional and fully connected layers. Among these models, BKVGG12 which consists of twelve layers, nine convolutional and three FC layers, showed the best accuracy rate of 71%. In [35], the authors detected the emotions with a single board computer, and their model was running on Intel's Movidius and Raspberry Pi. Moreover, [35] compared the performances of several architectures considering different numbers of layers and epochs; the highest validation accuracy was 67.79% with 406 epochs.

Study [36] analyzed facial expressions using the single-shot multibox detector (SSD) framework as a face detector and a pre-trained ResNet10 model. Also, this paper compares the performance of two models with 5 layers and one model with 7 layers; where the 5-layer models have 5 convolutional, 2 max-pooling, and 3 dropout layers, the 7-layer model has 7 convolutional, 2 max-pooling, and 3 dropout layers; the difference between 5-layer_1 and 5-layer_2 is batch normalization. All models used the AdaDelta optimizer with batch sizes of 128 and 200 epochs. The achieved accuracies of 5-layer_1, 5-layer_2 and 7_layer are, 62.2%, 62.4% and 59.2%, respectively. In [37], the authors proposed an end-to-end deep neural network based on CNN to address the FER issue. The results obtained with VGG plus Support Vector Machines (SVM) and GoogleNet models on the FER2013 dataset were 66.3% and 65.2%, respectively. However, the proposed model outperformed the others with an accuracy of 70%. Paper [37] emphasizes that a CNN architecture with less than 10 layers shows better results.

The authors of [38] proposed a new architecture called the "18-layer plain model" consisting of 5 blocks, inspired by VGG, which reached an accuracy of 69.21%. Afterward, they created a Multi Level Connection plain model (MLCNN), which reached a performance of 73.03%. Paper [39] used classical neural networks such as ResNet18, VGG16, AlexNet and compared their accuracies. All models utilized SGD optimizer with momentum 0, 9, batch size 64, and dropouts, in order to avoid overfitting. Moreover, the authors of [39] used the FER2013 dataset in two ways: the original dataset and the cropped one, with faces cut according to face landmarks. The achieved accuracy was 63.6%, 64.7%, and 68% with AlexNet, VGG16, ResNet, respectively, on the original dataset and 66.7%, 69.4%, and 70.7%, respectively, on the cropped one.

3 Materials and Methods

3.1 Dataset

There are many different databases to train automatic FER systems. Among the most used ones, the Japanese Female Facial Expressions (JAFFE) [40] is the oldest one, it stores 213 of 10 different Japanese female models; the Extended Cohn–Kanade (CK+) [41] collects 593 video sequences from 123 subjects, every video starts with a neutral expression and ends with a peak one; MMI database gathers 213 labeled sequences from 32 subjects, mainly in frontal pose [42,43]. However, all emotional frames of those databases are taken in laboratory conditions, and, therefore, they do not reflect the complexity of the real world. Recently, new databases of FER have been introduced with images that are collected from "the wild", i.e., real-world images with several disturbance elements such as rotation, occlusion, and light. FER2013 dataset [44] is one of the most realistic and challenging databases of FER currently present and, because of this, it has been used in this study. The FER2013 dataset includes 48×48 pixel grayscale images of faces with 6 different types of expressions plus the neutral pose. The distribution of emotions is as follows Angry (4,953), Disgust (547), Fear (5,121), Happy (8,989), Sad (6,077), Surprise (4,002), and Neutral (6,198). While the train set has 28.709 samples, the public test set consists of 3.589 emotional faces and the private test set includes 3.589 samples. The training dataset includes two columns: emotion and pixels. The emotion column contains

facial expressions in a numeric code ranging from 0 to 6 (0 = Angry, 1 = Disgust, 2 = Fear, 3 = Happy, 4 = Sad, 5 = Surprise, 6 = Neutral). Pixels are represented by strings that have space-separated pixel values in row-major order. The three main challenges of FER2013 can be listed as data imbalance, wrong labeled images, and poses with different angles. Considering the first challenge, the disgust emotion has only 547 images, which makes it harder to classify; this issue has been tackled using data augmentation. The second challenge is wrong labeled images; that is, FER2013 includes images with wrong labels. The last issue is that FER2013 includes faces viewed from different angles, i.e., there are pictures in which only one side of the faces is visible. These problems have a negative impact on classification.

3.2 Convolutional Neural Networks (CNN) Algorithms

In this study, several variations of Convolutional Neural Networks (CNN) architectures are used to challenge the FER2013 dataset. The success rates are compared, and the most successful model is adapted by two different robots. Recently, CNN architectures are very popular due to their remarkable achievements. A CNN consists of an input layer, an output layer, and a hidden layer that comprises several convolutional, pooling, fully connected, and/or normalization layers [45]. That is, CNN processes the image through numerous layers and successfully outputs the objects and their properties to the user. To date, the state-of-the-art in image classification, object detection, and image segmentation is reached by CNN [46]. The number and type of hidden layers allow the construction of different models. The LeNet [47] model is known as the starting point of CNN. LeNet achieved its first successful result in 1998. Experiments are conducted on the Modified National Institute of Standards and Technology (MNIST) dataset. LeNet architecture consists of 3 convolutional layers, 2 pooling layers, and 2 fully connected layers. AlexNet [48] is one of the most important studies in the field of CNN and deep learning which became remarkable in 2012. It includes 5 convolution layers and 3 fully connected layers. It has important features for training and local normalization that positively affects training speed. Moreover, AlexNet includes dropout and data augmentation which allows for solving the overfitting issue caused by large datasets with high-resolution [49]. ResNet [50] is a model that uses residual blocks with multiple layers to reduce training errors. ResNets have a variable number of layers that are coded in the ending number, such as ResNet18, ResNet34, ResNet50, ResNet101, and ResNet152. Residual block is the building block of a ResNet. A residual block includes convolution, activation (ReLU), and batch normalization. According to layer inputs, residual functions can be learned by residual blocks [51]. The big advantage of ResNet is to avoid the vanishing gradient problem [52]. VGGNet is another CNN architecture designed by Simonyan et al. of the Oxford Robotics Institute in 2014 [53]. VGGNet also has a variable number of layers. All variations include convolution layers, max-pooling layers, and fully connected layers. In this paper, VGG16 and VGG19 will be employed; the difference between VGG16 and VGG19 is in the number of convolutional layers. To achieve greater results, a variety of attempts have been made to upgrade AlexNet and compared to VGGNet; AlexNet includes more unrelated data in the last convolution layer, which affects the results [54]. Xception architecture is an extension of the Inception architecture [55]. Xception works with depthwise separable convolution layers instead of the standard Inception modules.

3.3 Robots

In order to test the proposed deep neural network algorithms, we developed two robots with different configurations. The first is based on the Nvidia Jetson Nano (NJN) development board which has 128 Cuda cores to run the variants of CNN. Moreover, the robot has been realized by connecting various sensors and components to its GPIO pins. In this implementation, a single Nano

board is sufficient for running the entire algorithm without the need for additional processing units. All operations and computing are carried out in a cutting-edge computing fashion. The second robot utilizes remote computing; therefore, there is no requirement for an additional graphical processing unit. It uses the NodeMCU development board to receive data from the remote computer and control the motion of the robot. Moreover, both robots provide the video stream remotely by configuring the Espressif ESP32 Cam and Raspberry Pi Zero (with Camera) development boards as IP cameras, respectively. Interesting to point out that the use of these wireless cameras, rather than the utilization of fixed cameras, dramatically increases flexibility during testing by allowing to gather images in various test conditions. For example, it is easy to remove the camera from the robot and get images from various heights and locations. Two robots can be seen in Fig. 1.

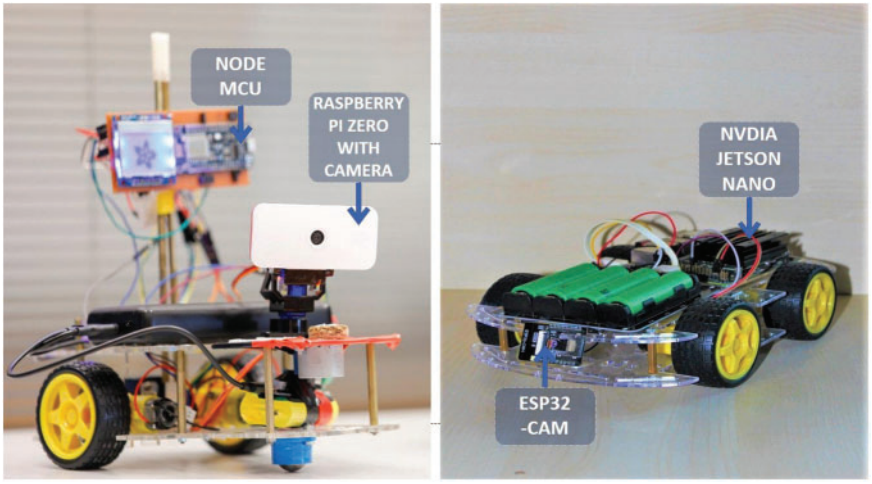

Figure 1: The two robots designed for testing the developed deep neural network algorithm: Remote computer-based robot (left), and Nvidia Jetson nano board-based robot (right)

3.4 Nvidia Jetson Nano Board Based Robot

The first robot is based on the Nvidia Jetson Nano (NJN) board. The system block diagram of the NJN-based robot is shown in Fig. 2. For the movement, we utilized four direct current (DC) motors that are connected to an L298N motor driver. The motor driver is connected to the general port input-output (GPIO) pins of NJN. ESP32 camera is configured as a WiFi IP camera, and it can be freely moved. Thus, during tests, we were able to use ESP32 Cam to perform the desired movements to the robot by showing various facial expressions. The process flow of the NJN robot is shown in Fig. 3.

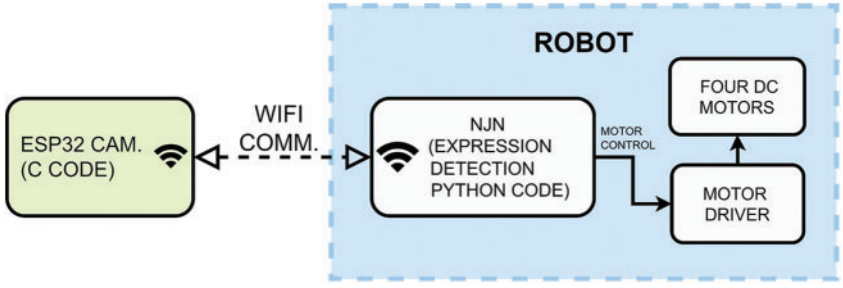

Figure 2: The system block diagram of the NJN board-based robot

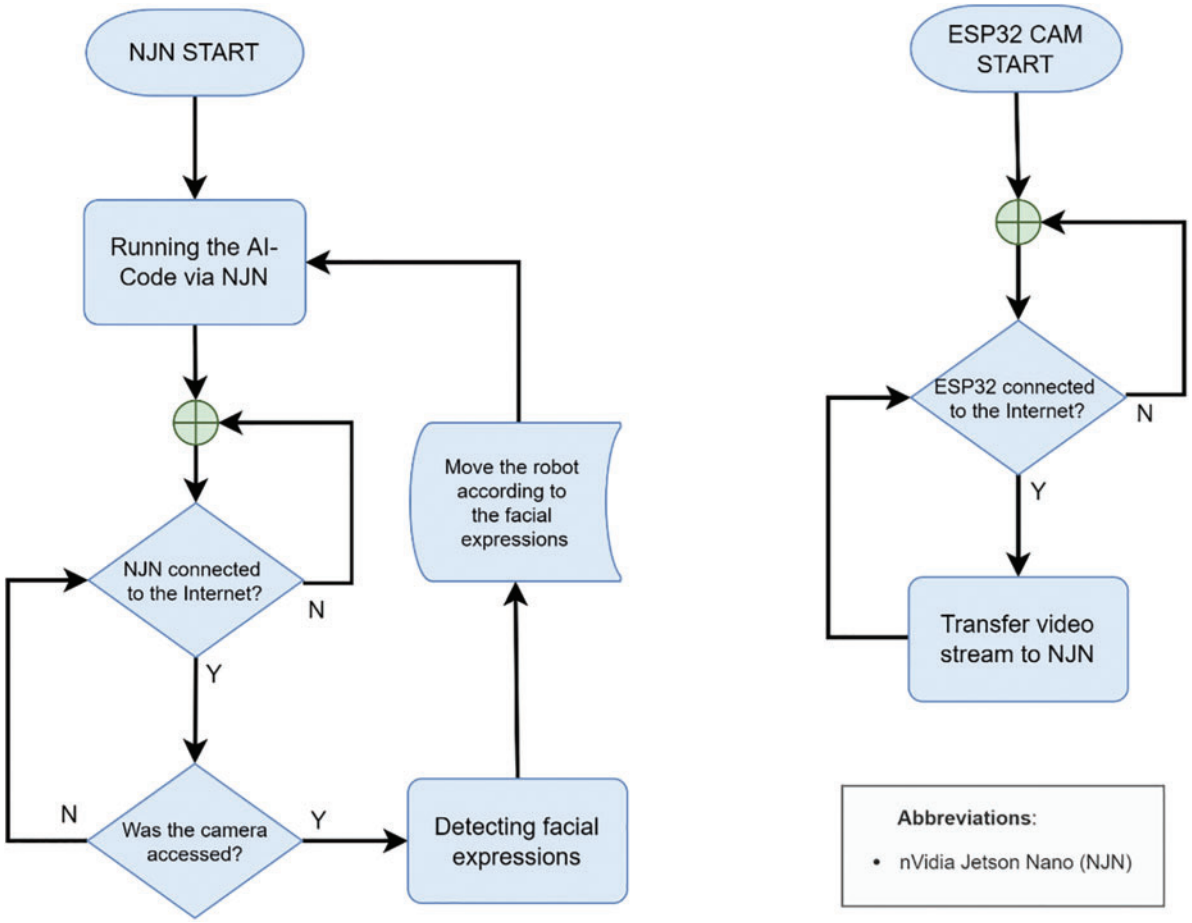

Figure 3: The process flow of the NJN board-based robot

In this configuration, our face recognition algorithm runs on NJN; the video stream from ESP32 CAM is processed on NJN. The robot moves according to the defined commands related to the facial expressions detected. In our implementation, the robot is programmed to detect facial expressions and perform the following actions:

1) Happy: The robot moves towards the person.
2) Sad: The robot moves slowly towards the human companion, timidly.
3) Angry: The robot moves back, quickly, as if running away from the person.
4) Surprised: The robot moves back and forth in short steps.
5) Neutral: The robot waits still.
6) Fear: The robot performs random short intensity movements forward, backward, left, and right; as it is feeling scared.
7) Disgust: In this case, the robot exhibits the movement of turning the head to the right and left in short intervals for mimicking a head shake of disapproval.

3.5 Remote Computer-Based Robot

In the second implementation, the algorithm runs on a remote computer that acts as a cloud-computing unit. The main advantage of the remote computing implementation is the high graphics

processing power provided by the GPU present in the server. Hence, the overall cost of the robot implementation is significantly reduced. In addition, the remote computing implementation enables controlling a high number of robots via the network. The configuration of the remote computer-based robot is shown in Fig. 4. In this setting, a Raspberry Pi development board configured as a WiFi IP camera gathers the video and transmits it to the remote computer. The robot performs tasks after the video stream is processed on the server using the algorithm. The movement of the robot is provided by dc motors connected to a NodeMCU board via an L298N motor driver. Two servo motors are also used to alter the perspective of the camera that is attached to the robot. The flow of the computing process of the remote computer-based robot is shown in Fig. 5. The server runs two programs; (1) the deep neural network Python program for facial expression detection, and (2) transferring data with the local C# server program to the port of the NodeMCU connected to the computer, controlling the robot with the NodeMCU and sending data. These two programs on the computer are communicating over the local server. The C# program consists of three main threads, which can be listed as (1) a local server thread, (2) a thread for preventing the server from collapsing due to overload, and (3) a thread for providing robot control. Firstly, the data is transferred to the local client thread in the CNN algorithms and then the data is transferred to the first thread in the C# code. Afterward, it is necessary to print these values captured in the second thread and created in the C# code, each second, in order to eliminate the server overload that will occur due to multiple speed data input. These values are printed from the second thread to the port where the NodeMcu Wifi development board is connected to the computer. In the third thread, the control and movement of the robot are provided. The control of the robot can also be carried out by the keyboard connected to the computer. Since facial expression recognition will not work properly while the robot is moving, firstly, the "G" key is used for selecting whether the facial expression detection or the robot control. In robot control mode, "W", "A", "S", and "D" keys are used to turn the robot back, forth, right, and left. In addition, the Pi IP camera on the robot can be positioned right and left up and down with two servo motors to which it is connected with the keys "1", "2", "3", "5". After setting the position of the robot and the viewpoint of the camera as desired, the camera mode is open again with the "F1" key. The control data and the detected expression information are written to the port where the NodeMCU development board is connected to the computer. The Server running on this card interacts with the Client on another NodeMCU card on the robot, the captured facial expression values are sent to the NodeMCU card on the robot and these values are printed on the LCD screen on the Robot. The remote computer also handles the synchronization of the overall system. It is based on checking the C# code, NodeMCU-Personal Computer (PC), and NodeMCU (Robot) code to check certain values to ensure the synchronization of the system. Before printing the C# code values (Detected Expressions), it checks if a simple command that NodeMCU (PC) has printed on the COM3 port has reached. Then, when the control process is completed, the C# code ensures that the desired data in the C# code is transferred to NodeMCU (PC). After that, this received data checks whether the client of NodeMCU (Robot) is connected to the NodeMCU (PC). The values are transferred to NodeMCU (Robot) if NodeMCU (Robot) client is available. All of this works in an endless loop and this prevents a possible crash in servers and clients. Fig. 4 shows the system block diagram of the remote computer-based agent, and Fig. 5 details its process flow.

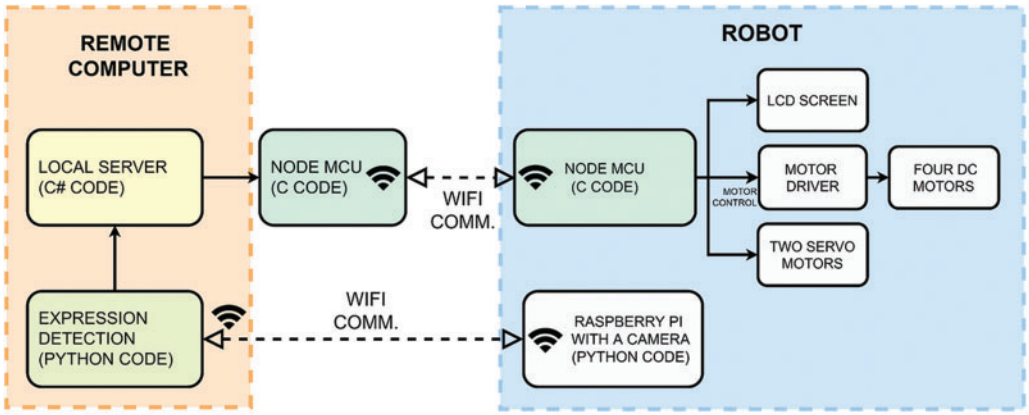

Figure 4: The system block diagram of the remote computer-based agent

3.6 Comparison of Robots

A comparison in terms of computing power and power dissipation of the two robot systems is given in Table 1. The remote computer-based robot detects the expressions and determines the related tasks (i.e., the movement to acknowledge the recognized emotion) faster due to the powerful CPU and GPU present in the system. The utilization of a remote computer for processing decreases the power consumption of the robot and increases its battery life of the robot. Overall, we concluded that the choice between two systems depends on the target application. For applications that require flexibility and slower detection performance the NJN system is suitable, however, for a crowded environment that requires a faster response and quick expression detection, the remote computer-based system should be preferred among these systems.

4 Results and Discussion

This study used different CNN models, such as VGG, AlexNet, Xception, ResNet, and LeNet plus the Ensemble method. The validation and test accuracies of all models are listed in Table 2. Overall, VGG16 is the most successful algorithm with a test accuracy of 70.8% while ResNet models showed the poorest performance, that is ResNet50 reaches only 58.8%. As shown in Table 2, there is a slight difference between the performance of VGG16 and VGG19; Fig. 6 plots the training and validation accuracies of both models.

Interesting to note that VGG16 includes the ReLU activation unit, which has a positive effect on the vanishing gradient problem. The higher accuracy was obtained using the dropout layer in VGG since dropout avoided over-fitting. Unfortunately, this option is not present in ResNet models.

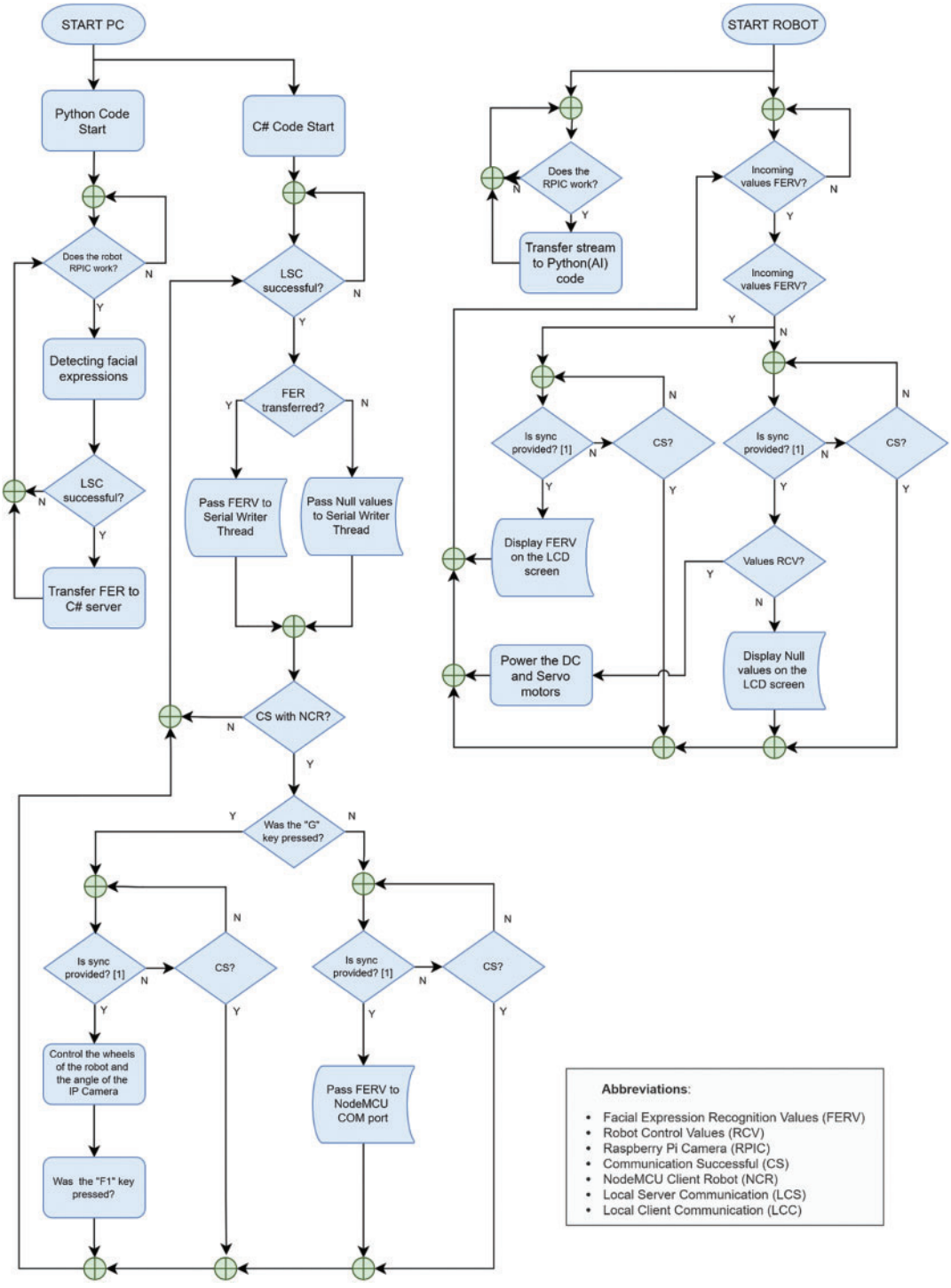

Figure 5: The process flow of the remote computer-based robot

Table 1: Comparison of the two robot systems

Hardware and parameter	NJN robot	Remote computer-based robot
CPU	ARM cortex-A573 quad core @1.48 GHz	Intel i7-7700HQ quad core @2.8 GHz
GPU	Maxwell-128 CUDA cores @922 MHz	1050TI (4 GB) @1.49 GHz
RAM	4 GB DDR4 @1600 MHz	16 GB DDR4 @2400 MHz
Average current dissipation	2.65 A (at 5 V)	1.3 A (at 5 V)
Average power dissipation	13.25 Watt	6.5 Watt (excluding dissipation of the remote computer)
Avg. working time (with 12000 mAH battery)	4.52 h	9.23 h

Table 2: Accuracy comparison of all models

Model name	Validation accuracy	Test accuracy
Ensemble	73.3%	75.1%
VGG16	70.4%	70.8%
VGG19	69.4%	69.6%
Xception	68.8%	68.2%
AlexNet	68.6%	69.6%
ResNet18	64.5%	64.3%
ResNet34	63.7%	63.4%
LeNet	59.8%	60.0%
ResNet50	56.5%	58.8%

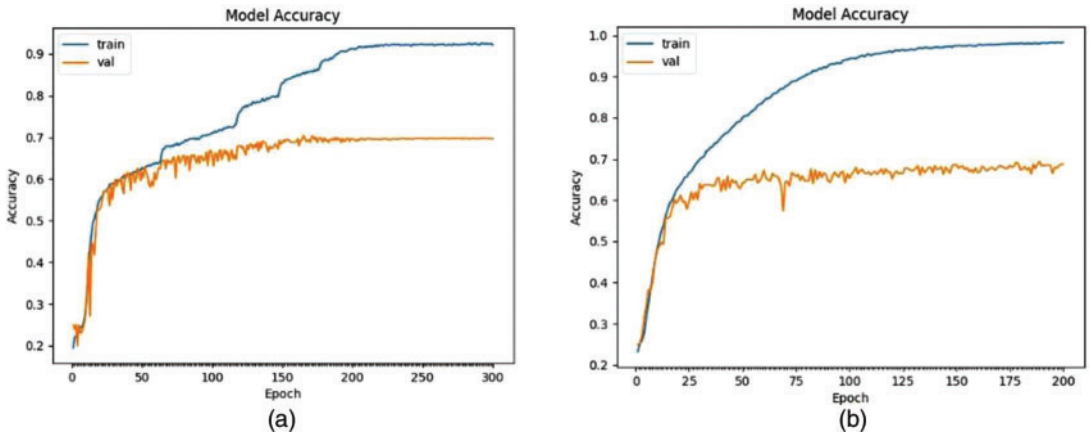

Figure 6: Plot of the model accuracy: (a) VGG16 and (b) VGG19

In summary, we tested different parameters such as many optimizers and learning rates, and VGG16 achieved the best result. We also employed an ensemble voting method, as [56], that uses seven different CNN models, such as VGG, AlexNet, ResNets, etc.; averaging of all methods has been used to combine the predictions of all models. Table 2 reports the performance of the ensemble method: 73.3% validation and 75.1% test accuracy. Finally, the ensemble soft voting technique improved validation and test accuracy by 2.9% and 4.3%, respectively. Fig. 7 shows the confusion matrix which has been created for the best model, VGG16, where rows label the actual classes and columns the predicted ones. Looking at Fig. 7, it is possible to infer that the lowest prediction is reached for the Scare expression (52%), and the best accuracy is achieved by the Happy one (90%); moreover, also Sad and Angry expressions have low recognition rate. As hyper-parameters, we used the SGD optimizer with momentum 0, 9, 300 epochs, 128 batch sizes, and different learning rates were implemented by using ReduceLROnPlateau, from 0.1 to 0.001. The original dataset has been augmented by producing synthetic data, using noise, cropping, scaling, translation, and rotation [57]. We utilized 300 epochs, a batch size of 128. According to the predicted errors, the model weights were updated together with the learning rate, as a high learning rate may lead to overfitting and oscillation. Important to underline that to get higher accuracy, losses should be minimized. An optimizer can decrease the loss function and improve accuracy. That is, by minimizing the function with optimizers, optimization and loss problems can be solved because the learning rate or different properties can be defined via optimizers. It is important which optimization technique is used to achieve the highest accuracy. Optimization algorithms such as SGD, Adagrad, Adadelta, Adam, and Adamax are widely used in many applications. There are differences between these algorithms in terms of performance and speed. In our study, SGD gives better results, when it is compared to Adam and RMSProp.

4.1 Qualitative Evaluation

Two surveys have been conducted among undergraduate students enrolled in a course offered by the Electrical and Electronics Department at Istanbul Bilgi University. After collecting basic information, such as age, sex, and previous experience with robots, both surveys had the following questions:

1) Would you like a robot to be an instructor in courses you take at your university?
2) Would you like a robot to be a student in your class at your university?
3) Would you like a robot to be a department or faculty secretary at your university?
4) Would you like a robot to be a janitor at your university?
5) Would you like a robot to be a security staff member at your university?
6) Would you like a robot to be a dean or rector at your university?

Possible answers were on a scale of 5: (1) Definitely yes, (2) Yes, (3) Neutral (not sure), (4) No, and (5) Absolutely no. The 1st survey has been attended by 55 (44 M, 11 F) undergraduates, 95% 20–25 years old and 5% 25–30 years old; 48% of them had previous practical experience in robotics. After the 1st survey, the undergraduates watched a video presenting the two robots interacting with the students, co-authors of the paper, who designed and built the first robot. The 2nd survey has been attended by 47 (38 M, 9 F) students, a subset of the original group, 94% aged 20–25 years and 6% 25–30 years old; 58% of them had previous practical experience in robotics. In the following, every question has been analyzed independently before and after the projection of the video. The answers are grouped into positive (YES), which is the sum of the percentages received by answers "Definitely yes" and "Yes", neutral, and negative (NO), which is the sum of the percentages received by "No" and "Absolutely no". Table 3 summarizes the results that are also plotted in Fig. 8 (the winner class in bracket). Looking at

the outputs of the surveys, it is interesting to notice that the original acceptance rate increased after the projection of the video for all roles, except for the instructor's role and the security staff for which there were no changes. In all cases, the winner class did not change in the 2nd survey, but the acceptance rate, generally, increased. Furthermore, the results highlight that the undergraduate students at Istanbul Bilgi University are quite ready to accept robots in the roles of student, janitor, secretary, and security staff, but they reject the idea of artificial companions as instructors or deans, or rectors. Finally, it is interesting to notice that the rejection of the dean and rector roles decreased by 10% after watching the video, in the 2nd survey.

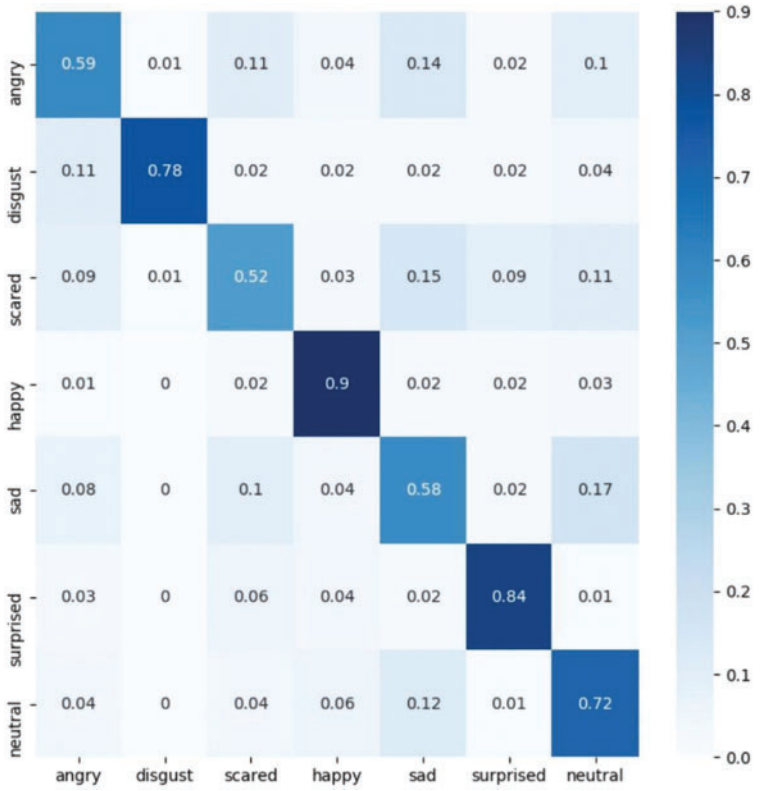

Figure 7: Confusion matrix of VGG16

Table 3: Results of the two surveys

	Winner class	Before (%)	After (%)
Instructor	No	43.6%	46.8%
Student	Yes	43.6%	46.8%
Secretary	Yes	58.2%	65.9%
Janitor	Yes	49.1%	53.1%
Security	Yes	54.5%	53.3%
Dean or rector	No	63.6%	53.2%

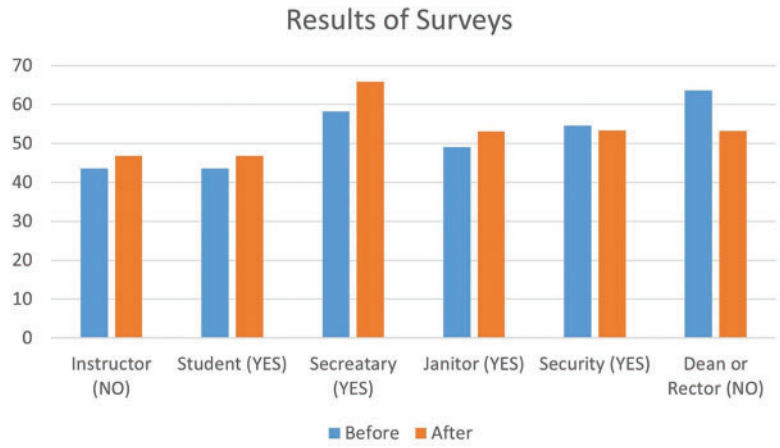

Figure 8: Comparison of the results of the two surveys

5 Conclusion

Robot companions will soon be part of our everyday life and students in the Engineering Faculty must be trained to design, build, and interact with them. Two affordable robots are designed and undergraduate students played with them, with the two-fold objective to trigger their mental models that affect acceptance of artificial companions and to test the proposed affective reaction of the robots. Since automatic FER is a necessary pre-processing step for acknowledgment of emotions, this paper tested different variations of CNN to detect six facial expressions plus the neutral face. We carried out various experiments and addressed different issues such as learning rate, epochs, data augmentation, batch size, and optimizer. The state-of-the-art performance of 75.1% on the FER2013 database has been reached by the ensemble voting method and VGG16 with 70.8%, the most successful model for facial expression detection, has been implemented into the robots. For testing the algorithm in different conditions, we designed two robots: (1) an Nvidia Jetson Nano (NJN) board robot and (2) a remote computer-based one. The model trained on the NJN development board has been programmed to perform certain actions according to the detected facial expressions. The second robot system is designed for testing the algorithm in a high-performance computing platform. With these experiments, we successfully demonstrated the feasibility of facial expression detection and acknowledging algorithms on both edge computing-based and remote computer-based robot systems. Finally, the surveys run in this work empirically prove the assumption that student-robot interaction helps to mature a positive attitude towards virtual agents.

Funding Statement: The authors received no specific funding for this study.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] R. R. Murphy, S. Tadokoro, D. Nardi, A. Jacoff, P. Fiorini *et al.*, "Search and rescue robotics," In: B. Siciliano and O. Khatib (Eds.), *Springer Handbook of Robotics*, pp. 1151–1173, Berlin Heidelberg: Springer, 2008.

[2] F. Feng, D. Li, J. Zhao and H. Sun, "Research of collaborative search and rescue system for photovoltaic mobile robot based on edge computing framework," in *Proc. Chinese Control and Decision Conf. (CCDC)*, Hefei, China, pp. 2337–2341, 2020.
[3] K. Nosirov, S. Begmatov and M. Arabboev, "Analog sensing and leap motion integrated remote controller for search and rescue robot system," in *Proc. Int. Conf. on Information Science and Communications Technologies (ICISCT)*, Tashkent, Uzbekistan, pp. 1–5, 2020.
[4] D. N. S. R. Kumar and D. Kumar, "VNC server based robot for military applications," in *Proc. IEEE Int. Conf. on Power, Control, Signals and Instrumentation Engineering (ICPCSI)*, Chennai, India, pp. 1292–1295, 2017.
[5] S. Joshi, A. Tondarkar, K. Solanke and R. Jagtap, "Surveillance robot for military application," *International Journal of Engineering and Computer Science*, vol. 7, no. 5, pp. 23939–23944, 2018.
[6] G. G. Hady, C. D. Abigail, H. Sebastian, D. Q. Nicola, A. Andrea *et al.*, "Alcides: A novel lunar mission concept study for the demonstration of enabling technologies in deep-space exploration and human-robots interaction," *Acta Astronautica*, vol. 151, pp. 270–283, 2018.
[7] Y. Huang, S. Wu, Z. Mu, X. Long, S. Chu *et al.*, "A Multi-agent reinforcement learning method for swarm robots in space collaborative exploration," in *Proc. 6th Int. Conf. on Control, Automation and Robotics (ICCAR)*, Singapur, pp. 139–144, 2020.
[8] F. Gul, I. Mir, W. Rahiman and T. Islam, "Novel implementation of multi-robot space exploration utilizing coordinated multi-robot exploration and frequency modified whale optimization algorithm," *IEEE Access*, vol. 9, no. 22, pp. 774–787, 2021.
[9] C. Lacey and C. Caudwell, "Cuteness as a 'dark pattern' in home robots," in *Proc. 14th ACM/IEEE Int. Conf. on Human-Robot Interaction (HRI)*, Daegu, Korea, pp. 374–381, 2019.
[10] J. A. Rincon, A. Costa, C. Carrascosa, P. Novais and V. Julian, "Emerald-exercise monitoring emotional assistant," *Sensors*, vol. 19, no. 8, pp. 1–21, 2019.
[11] L. Zhijie, C. Mingyu, Z. Zhiming and Y. Youling, "Design and implementation of home service robot," in *Proc. Chinese Automation Congress (CAC)*, Shanghai, China, pp. 3541–3546, 2020.
[12] M. Zhang, G. Tian, Y. Zhang and P. Duan, "Service skill improvement for home robots: Autonomous generation of action sequence based on reinforcement learning," *Knowledge-Based Systems*, vol. 212, pp. 106605, 2021.
[13] H. Huttenrauch and K. S. Eklundh, "Fetch-and-carry with cero: Observations from a long-term user study with a service robot," in *Proc. 11th IEEE Int. Workshop on Robot and Human Interactive Communication*, Berlin, Germany, pp. 158–163, 2002.
[14] M. Abdul Kader, M. Z. Islam, J. Al Rafi, M. Rasedul Islam and F. Sharif Hossain, "Line following autonomous office assistant robot with PID algorithm," in *Proc. Int. Conf. on Innovations in Science, Engineering and Technology (ICISET)*, Chittagong, Bangladesh, pp. 109–114, 2018.
[15] T. Kanda, M. Shimada and S. Koizumi, "Children learning with a social robot," in *Proc. 7th ACM IEEE Int. Conf. on Human—Robot Interaction (HRI)*, Boston, USA, pp. 351–358, 2012.
[16] L. Brown, R. Kerwin and A. M. Howard, "Applying behavioral strategies for student engagement using a robotic educational agent," in *Proc. IEEE Int. Conf. on Systems, Man and Cybernetics*, Manchester, UK, pp. 4360–4365, 2013.
[17] G. Gordon, C. Breazeal and S. Engel, "Can children catch curiosity from a social robot?" in *Proc. the 10th Annual ACM/IEEE Int. Conf. on Human-Robot Interaction (HRI'15)*, Portland, Oregon, USA, pp. 91–98, 2015.
[18] J. Michaelis and B. Mutlu, "Supporting interest in science learning with a social robot," in *Proc. Interaction Design and Children (IDC)*, Boise, USA, pp. 71–82, 2019.
[19] H. G. Okuno, K. Nakadai and H. Kitano, "Social interaction of humanoid robot based on audio-visual tracking," In: T. Hendtlass and M. Ali (Eds.), *Developments in Applied Artificial Intelligence, Lecture Notes in Computer Science*, vol. 2358, pp. 725–735, Berlin, Heidelberg: Springer, 2002.

[20] S. Sabanovic, M. P. Michalowski and R. Simmons, "Robots in the wild: Observing human-robot social interaction outside the lab," in *Proc. 9th IEEE Int. Workshop on Advanced Motion Control*, İstanbul, Turkey, pp. 596–601, 2006.
[21] J. Pineau, M. Montemerlo, M. Pollack, N. Roy and S. Thrun, "Towards robotic assistants in nursing homes: Challenges and results," *Robotics and Autonomous Systems: Special Issue on Socially Interactive Robots*, vol. 42, no. 3, pp. 271–281, 2003.
[22] D. Fischinger, P. Einramhof, K. E. Papoutsakis, W. Wohlkinger, P. Mayer *et al.*, "Hobbit, a care robot supporting independent living at home: First prototype and lessons learned," *Robotics and Autonomous Systems*, vol. 75, pp. 60–78, 2016.
[23] F. K. Z. Petric, "Design and validation of MOMDP models for child–robot interaction within tasks of robot assisted ASD diagnostic protocol," *International Journal of Social Robotics*, vol. 12, pp. 371–388, 2020.
[24] D. Vogiatzis, C. Spyropoulos, S. Konstantopoulos, V. Karkaletsis, Z. Kasap *et al.*, "An affective robot guide to museums," in *Proc. 4th Int. Workshop on Human-Computer Conversation*, Bellagio, Italy, 2008.
[25] B. Han, Y. Kim, K. Cho and H. S. Yang, "Museum tour guide robot with augmented reality," in *Proc. 16th Int. Conf. on Virtual Systems and Multimedia*, Seoul, Korea, pp. 223–229, 2010.
[26] B. P. E. A. Vasquez and F. Matia, "A Tour-guide robot: Moving towards interaction with humans," *Engineering Applications of Artificial Intelligence*, vol. 88, pp. 103356, 2020.
[27] B. Ruyter, P. Saini, P. Markopoulos and A. Breemen, "Assessing the effects of building social intelligence in a robotic interface for the home," *Interacting with Computers*, vol. 17, no. 5, pp. 522–541, 2005.
[28] K. Dautenhahn, "Socially intelligent robots: Dimensions of human-robot interaction," *Philosophical Transactions of the Royal Society of London. Series B, Biological Sciences*, vol. 362, no. 1480, pp. 679–704, 2007.
[29] C. Hieida, T. Horii and T. Nagai, "Toward empathic communication: Emotion differentiation via face-to-face interaction in generative model of emotion," in *Proc. Joint IEEE 8th Int. Conf. on Development and Learning and Epigenetic Robotics*, Tokyo, Japan, pp. 66–71, 2018.
[30] G. J. Hofstede, "Grasp agents: Social first, intelligent later," *AI & Society*, vol. 34, no. 1, pp. 535–543, 2019.
[31] U. Gerecke and B. Wagner, "The challenges and benefits of using robots in higher education," *Intelligent Automation Soft Computing*, vol. 13, pp. 29–43, 2007.
[32] C. Pramerdorfer and M. Kampel, "Facial expression recognition using convolutional neural networks: State of the art," arXiv Preprint arXiv:1612.02903, 2016.
[33] A. Mollahosseini, D. Chan and M. H. Mahoor, "Going deeper in facial expression recognition using deep neural networks," in *Proc. IEEE Winter Conf. on Applications of Computer Vision (WACV)*, Lake Placid, NY, USA, pp. 1–10, 2016.
[34] D. V. Sang, N. Van Dat and D. P. Thuan, "Facial expression recognition using deep convolutional neural networks," in *Proc. 9th Int. Conf. on Knowledge and Systems Engineering (KSE)*, Hue, Vietnam, pp. 130–135, 2017.
[35] V. Srinivasan, S. Meudt and F. Schwenker, "Deep learning algorithms for emotion recognition on low power single board computers," in *Proc. IAPR Workshop on Multimodal Pattern Recognition of Social Signals in Human Computer Interaction*, Stockholm, Sweden, pp. 59–70, 2018.
[36] M. Jangid, P. Paharia and S. Srivastava, "Video-based facial expression recognition using a deep learning approach," In: S. Bhatia, S. Tiwari, K. Mishra and M. Trivedi (Eds.), *Advances in Computer Communication and Computational Sciences. Advances in Intelligent Systems and Computing*, vol. 924, pp. 653–660, Singapore: Springer, 2019.
[37] S. Minaee and A. Abdolrashidi, "Deep-emotion: Facial expression recognition using attentional convolutional network," arXiv Preprint arXiv:1902.01019, 2019.
[38] D. H. Nguyen, S. Kim, G. Lee, H. Yang, I. Na *et al.*, "Facial expression recognition using a temporal ensemble of multi-level convolutional neural networks," *IEEE Transactions on Affective Computing*, vol. 13, no. 1, pp. 226–237, 2022.

[39] G. Shengtao, X. Chao and F. Bo, "Facial expression recognition based on global and local feature fusion with CNNs," in *Proc. IEEE Int. Conf. on Signal Processing, Communications and Computing (ICSPCC)*, Dalian, China, pp. 1–5, 2019.
[40] M. Lyons, S. Akamatsu, M. Kamachi and J. Gyoba, "Coding facial expressions with gabor wavelets," in *Proc. Third IEEE Int. Conf. on Automatic Face and Gesture Recognition*, Nara, Japan, pp. 200–205, 1998.
[41] P. Lucey, J. Cohn, T. Kanade, J. Saragih, Z. Ambadar *et al.*, "The extended cohn-kanade dataset (CK+): A complete dataset for action unit and emotion-specified expression," in *Proc. IEEE Computer Society Conf. on Computer Vision and Pattern Recognition—Workshops*, San Francisco, CA, USA, pp. 94–101, 2010.
[42] M. Pantic, M. Valstar, R. Rademaker and L. Maat, "Web-based database for facial expression analysis," in *Proc. IEEE Int. Conf. on Multimedia and Expo*, Amsterdam, Netherlands, pp. 5, 2005.
[43] M. Valstar and M. Pantic, "Induced disgust, happiness and surprise: An addition to the mmi facial expression database," in *Proc. Int. Conf. Language Resources and Evaluation, Workshop Emotion*, Malta, pp. 65–70, 2010.
[44] I. J. Goodfellow, D. Erhan, P. L. Carrier, A. Courville, M. Mirza *et al.*, "Challenges in representation learning: A report on three machine learning contests," In: M. Lee, A. Hirose, Z. G. Hou and R. M. Kil (Eds.), *Neural Information Processing. ICONIP 2013. Lecture Notes in Computer Science*, vol. 8228, pp. 117–124, Berlin: Springer, 2013.
[45] K. O'Shea and R. Nash, "An introduction to convolutional neural networks," arXiv Preprint arXiv:1511.08458, 2015.
[46] Z. Q. Zhao, P. Zheng, S. T. Xu and X. Wu, "Object detection with deep learning: A review," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 30, no. 11, pp. 3212–3232, 2019.
[47] Y. LeCun, L. Bottou, Y. Bengio and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.
[48] A. Krizhevsky, I. Sutskever and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Advances in Neural Information Processing Systems*, vol. 25, pp. 1097–1105, 2012.
[49] C. Shorten and T. M. Khoshgoftaar, "A survey on image data augmentation for deep learning," *Journal of Big Data*, vol. 6, no. 1, pp. 1–48, 2019.
[50] K. He, X. Zhang, S. Ren and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, Las Vegas, USA, pp. 770–778, 2016.
[51] A. Shah, E. Kadam, H. Shah, S. Shinde and S. Shingade, "Deep residual networks with exponential linear unit," in *Proc. Third Int. Symp. on Computer Vision and the Internet*, New York, USA, pp. 59–65, 2016.
[52] M. A. Nielsen, "The vanishing gradient problem neural networks and deep learning," *The Neural Networks and Deep Learning*, pp. 151–164, SF, USA: Determination Press, 2015. [Online]. Availalble at: https://neuralnetworksanddeeplearning.com
[53] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," arXiv Preprint arXiv:1409.1556, 2014.
[54] W. Yu, K. Yang, Y. Bai, H. Yao and Y. Rui, "Visualizing and comparing convolutional neural networks," arXiv Preprint arXiv:1412.6631, 2014.
[55] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, Honolulu, USA, pp. 1251–1258, 2017.
[56] A. Khanzada, C. Bai and F. T. Celepcikay, "Facial expression recognition with deep learning," arXiv Preprint arXiv:2004.11823, 2020.
[57] A. M. lajczyk and M. Grochowski, "Data augmentation for improving deep learning in image classification problem," in *Proc. Int. Interdisciplinary PhD Workshop (IIPhDW)*, Świnoujście, Poland, pp. 117–122, 2018.