Facial Expression Recognition to Detect Student Engagement in Online Lectures

Joko Siswantoro[1]*, Januar Rahmadiarto[2], Mohammad Farid Naufal[3]

[1,2,3]Department of Informatics, University of Surabaya, Surabaya, East Java, Indonesia
Email: [1*]joko_siswantoro@staff.ubaya.ac.id, [2]s160418042@student.ubaya.ac.id, [3]faridnaufal@staff.ubaya.ac.id

(Received: 26 May 2024, revised: 10 Jun 2024, accepted: 11 Jun 2024)

Abstract

In synchronous online lectures, the lecturers often provide the lecture material directly through video conference technology. On the other hand, there are many students who do not pay attention to the lecturers when they are participating in online lectures. As a consequence, in this research, an application was developed to assist lecturers in gathering data regarding the degree to which students who participate in online lectures pay attention to the presented information. The application employed a convolutional neural network (CNN) model to recognize each student's facial expressions and place them into one of two classes: either engaged or disengaged. The captured student facial image was preprocessed to facilitate the classification process. The preprocessing stage consisted of image conversion to gray scale, face detection using the Haar-Cascade Classifier model, and a median filter to reduce noise. In the process of designing a CNN model, three different hyperparameter tuning scenarios were implemented. These tuning scenarios aimed to obtain the best possible CNN model by determining which CNN model hyperparameters were the most optimal. The results of the experiments indicate that the CNN model from the second scenario has the highest level of accuracy in terms of recognizing facial expressions, coming in at 86%. The results of this research have been tested to measure the level of student participation in online lectures. The trial results show that the proposed application can help lecturers evaluate student engagement during online lectures.

Keywords: Facial Expression Recognition, Convolutional Neural Network, Student Engagement, Online Lectures.

I. INTRODUCTION

Online lectures are learning processes using virtual classrooms on the Internet [1], [2]. Online lectures have become indispensable in various educational institutions since the Covid-19 pandemic [3]. But after the Covid-19 pandemic passed, online lectures still have an important role in education [4]. Two popular methods are used in conducting online lectures, namely synchronous and asynchronous. In the synchronous method, lectures are delivered directly by lecturers to students through video conferencing applications. In addition, lecturers and students can also meet face-to-face and communicate directly in virtual classrooms. Students can also work together virtually at the same time. In the asynchronous method, lecturers usually provide learning videos or written learning materials in files uploaded to the learning site so that students can access and study at a specified time. Communication between lecturers and students is usually done by sending messages through discussion boards available on learning sites [5].

The effectiveness of online lectures is still very low compared to offline face-to-face lectures. This is the impact of low student engagement in the respective online lectures. Students are only present in the virtual classroom but then do other activities unrelated to the lecture and do not pay attention to the lecture's material [6]. On the other hand, lecturers need to evaluate student engagement in a lecture activity because student engagement is one important aspect of determining student success in a lecture, especially online lectures [7].

In online lectures, it is difficult for lecturers to see student engagement directly, even though students are asked to turn on the camera when explaining synchronous online lecture material. This can happen because the lecturer focuses more on the materials delivered. Therefore, an application is needed to assist lecturers in evaluating student engagement during synchronous online lectures. Some methods that can be used to develop such applications are recording brain and heart signals [8], measuring heart rate [9], using context-performance [10], and facial expression recognition [11]. Due to the ease of obtaining and using digital cameras, facial expression recognition is the most widely used method to detect student engagement in lectures [12].

Several researchers have developed various methods to recognize participants' facial expressions in both online and offline learning classes. Chen et al. [13] proposed the use of support vector machines (SVM) [14] to recognize facial expressions in e-learning. Two feature extraction methods, namely facial shape features [15] and Gabor wavelets [16], are used to extract features from face images. The experiments conducted on the Japanese Female Facial Expression (JAFFE) dataset [17] found that the facial shape feature produces better facial expression recognition accuracy, which is above 80% when compared to the Gabor wavelets feature. However, the exact value of accuracy was not reported by the author. Whitehill et al. [18] proposed an approach to recognize student engagement from facial expressions automatically. Three classification methods were used to recognize students' facial expressions in the proposed approach: GentleBost [19], SVM, and multinomial logistic regression [20]. The best recognition accuracy achieved was only 72.9% using SVM and Gabor features on the HCBU dataset [21].

In addition to using machine learning models to recognize facial expressions in detecting student engagement in lectures, some researchers also use deep learning models, as reported in [12], [22], [23]. Nezami et al. [12] proposed the use of a convolutional neural network (CNN) model based on the VGG-B model [24]. In the initial stage, the VGG-B model was trained using the facial expression recognition 2013 (FER-2013) dataset [25] to initialize the model weight values. Furthermore, the VGG-B model was trained using the engagement recognition (ER) dataset created in the study to classify student face images into two classes engaged and disengaged. Experimental results showed that the proposed model only achieved 72.38% accuracy in recognizing student engagement in lectures. Pabba et al. [22] proposed an intelligent system to monitor student engagement in offline lectures in large classes based on facial expressions in real-time. A CNN model that adapted the VGGNet model [24] was trained to classify students' facial expressions into six classes. Although it could simultaneously assess student engagement, the proposed system only achieved 76.90% accuracy.

Classification models proposed to recognize student facial expressions were almost all trained using foreign face image datasets. Therefore, these models may not be suitable for recognizing the facial expressions of Indonesians. Although some of the proposed systems have used a pre-trained CNN model, the accuracy achieved is still below 80%. This can happen because when modifying the fully connected layer of the CNN model, the researchers did not optimize the number of dense layers and the number of neurons in each dense layer. In addition, almost all of the proposed systems can only recognize the facial expressions of one student at a time, except for the system reported in [22]. Nurdiati et al. [23] compared two CNN model architectures, namely AlexNet [26] and VGG [24], for the facial expression recognition of Indonesian students in one class. Both models will classify students' facial expressions into three classes smile, grimace, and neutral. The experimental results show that the Alexnet model produced better accuracy than the VGG, which is 100%. However, this study only involved seven student faces and was not used to recognize student engagement in lectures.

Several pre-trained CNN models have been used to recognize facial expressions from images. However, the performance of these models varies from case to case. In addition, these models have not been applied to detect Indonesian students' engagement in lectures both online and offline. Therefore, in this research, an application will be developed to detect student engagement in online lectures based on facial expression recognition. A CNN model will be trained to classify students' facial expressions into two classes: paying attention and not paying attention. The best CNN model architecture will be determined in experiments by performing three hyperparameter tuning scenarios. The developed application will receive input images of students' faces attending online lectures. The image is then processed, and each face in the image will be classified using a CNN model that has been trained to detect student engagement in lectures.

II. MATERIAL AND METHOD

In this research, six main steps were carried out, namely dataset acquisition, image preprocessing, CNN model development, model training, model evaluation, and CNN model implementation on the application to detect student involvement in lectures. The flow of the steps of this research can be seen in Figure 1.

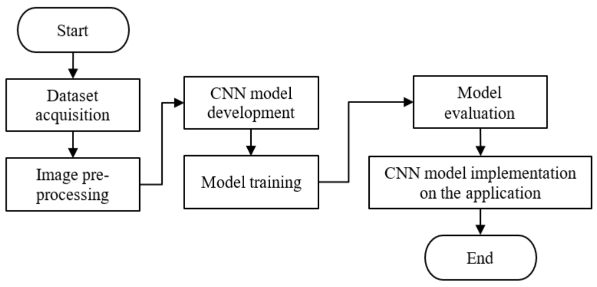

Figure 1. The Flow of Research Steps.

A. Dataset Acquisition

The acquisition of the student facial expression images dataset involved 320 student volunteers who were asked to participate in an online lecture through the Zoom application. All volunteers were asked to turn on their respective cameras during the lecture. A total of 180 volunteers were asked to pay serious attention to the lecture. The remaining 140 volunteers were asked to be sleepy, daydreaming, watching videos, playing games, and talking to friends. The face images of all volunteers were then captured in RGB (red, green, blue) color space and saved into a file. Each facial image was then labeled as engaged or disengaged according to each volunteer’s activity during the online lecture. Some examples of facial expression images labeled as engaged and disengaged can be seen in Figure 2 and Figure 3, respectively. The Authors have received permission from the student volunteers whose face images were used in this research to publish his/her face images in this article.

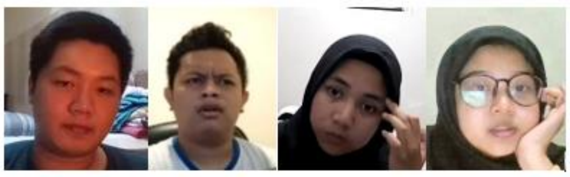
Figure 2. Example of a Facial Expression Image with Label Engaged.

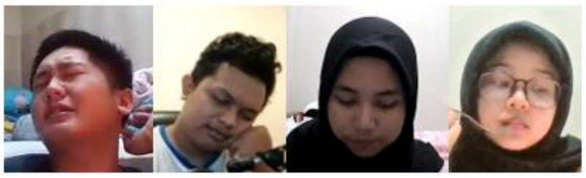
Figure 3. Example of a Facial Expression Image with Label Disengaged.

B. Image Preprocessing

Image preprocessing is a process for processing captured images to be used in the classification process and produce good classification accuracy. In this research, the image pre-processing performed converts the RGB image to grayscale, face area detection, resizes the image, and reduces noise. Converting an RGB image to a grayscale image involves converting the color intensity value of the RGB image at each pixel into a single value that represents the gray level by calculating the weighted average of the red ®, green (G), and blue (B) color intensity values at each pixel of the RGB image as in Eq. (1), and replacing the color intensity value at that pixel with the average value [27].

$$grayscale\ value = 0.2126R + 0.7152G + 0.0722B \quad (1)$$

Face area detection was performed using the Haar-Cascade Classifier model, a method for object detection in images based on Haar-like features [19]. Haar-Cascade Classifier is one of the effective object detection techniques and has been widely used to detect face areas in images. The model has been trained using many examples of face and non-face images to learn patterns of features typical of faces, such as eye, nose, and mouth lines. The model is then used to detect face areas in grayscale images by identifying matching Haar-like feature patterns. The grayscale image was then cropped according to the detected face area.

A median filter was applied to the face image before input to the CNN model to reduce noise in the image. This filter works by replacing the intensity value of a pixel with the median value of the intensity of neighboring pixels [27]. This study employed a median filter with a kernel size of 5×5. Figure 4 shows an example of the results of the image preprocessing step.

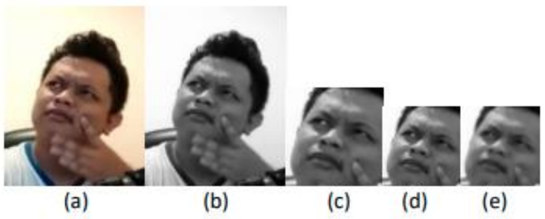

Figure 4. Examples of Image Preprocessing Results: (a) Original Image, (b) Grayscale Image, (c) Face Detection Result, (d) Resized Result, (e) Median Filter Result.

C. CNN Model Development

This research developed a CNN architecture from scratch consisting of convolutional layers, max pooling layers, and fully connected layers ending with a softmax layer for classification. Convolution layer and pooling layer used to extract features from images and reduce their dimensions through sub-sampling layers to reduce computational complexity and improve computational efficiency.

To obtain optimum classification accuracy, hyperparameter tuning was carried out on the CNN model with three scenarios using the random search algorithm [28]. Hyperparameter tuning using the random search algorithm is essential to efficiently exploring a wide range of hyperparameter combinations and identifying the optimal settings for the CNN model. This approach enhances classification accuracy by systematically optimizing critical hyperparameters, ensuring the model performs at its best across various scenarios. In the first scenario, tuning was performed on two hyperparameters: the number of filters in the convolutional layer with a search domain in $[32{,}512]$ and the number of nodes in the fully connected layer with a search domain in $[128{,}512]$. All filters in the convolutional layer use a 3×3 kernel and ReLU (Rectified Linear Unit) activation function as defined in Eq. (2).

$$ReLU(x) = \max(0, x) \quad (2)$$

In the second scenario, tuning was performed on three hyperparameters: the number of filters in the convolutional layer, the number of nodes in the fully connected layer, and the size of the filter kernel in the convolutional layer. The search domains for each hyperparameter were in $[32{,}512]$, $[128{,}512]$, and $[32{,}512]$, respectively. All convolutional layers used ReLU activation function in this scenario.

In the third scenario, four hyperparameters were tuned: the number of filters in the convolutional layer, the number of nodes in the fully connected layer, the size of the filter kernel in the convolutional layer, and the activation function in the convolutional layer. The search domains for each hyperparameter were in $[32{,}512]$, $[128{,}512]$, $\{1 \times 1, 3 \times 3\}$, and $[32{,}512]$, respectively. The sigmoid and tanh functions are defined as in Eq. (3) and Eq. (4), respectively.

$$S(x) = \frac{1}{1 + e^{-x}} \quad (3)$$

$$tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \quad (4)$$

Each scenario employed three convolutional layers, one fully connected layer, and one output layer with two nodes and a softmax activation function, as in Eq. (5). Each convolutional layer was followed by a max pooling layer with a kernel size of 2 × 2. The output of the last max pooling layer was then converted into a 1-dimensional tensor before entering the fully connected layer. Furthermore, 20 models were generated by the random search algorithm to find the best hyperparameter combination in each scenario. Details of the CNN model architecture can be seen in Table 1.

$$\sigma(\mathbf{x})_i = \frac{e^{x_i}}{\sum_{j=1}^{n} e^{x_j}}, \mathbf{x} = x_1, x_2, \dots, x_n \quad (5)$$

Table 1. CNN Model Architecture Details and Hyper-Parameter Values for Each Scenario.

Layer	N	Hyper-parameter	Search domains		
			S_1	S_2	S_3
Convo-lutional	3	Filter's number	32-512	32-512	32-512
		Kernel size	3 × 3	1 × 1, 3 × 3	1 × 1, 3 × 3
		Activation function	ReLU	ReLU	ReLU, sigmoid, tanh
Max pooling	3	Kernel size	2 × 2	2 × 2	2 × 2
Fully con-nected	1	Node size	128-512	128-512	128-512
		Activation function	ReLU	ReLU	ReLU, sigmo-id, tanh
Output	1	Node size	2	2	2
		Activation function	Soft-max	Soft-max	Soft-max

N: layer size, S_1: Scenario 1, S_2: Scenario 2, S_3: Scenario 3

D. Model Training

The image dataset created was divided into two mutually exclusive subsets, namely training data and testing data, with a ratio of 80:20. The training data was used to train all layers of the CNN model in all scenarios using Adam [29] as the training algorithm. Each model was trained with 20 epochs and a sample size of 32 data at each iteration. In this study, the Adam training algorithm used a learning rate of 0.001 and a momentum of 0.99. The objective function used in the training process was categorical cross entropy, as in Eq. (6),

$$L(\mathbf{y}, \hat{\mathbf{y}}) = -\sum_{i=1}^{K} y_i \log \hat{y}_i \quad (6)$$

where K is the sample size, $\mathbf{y} = (y_1, y_2, \dots, y_K)$ is the one-hot encoding vector of the actual class labels, and $\hat{\mathbf{y}} = (\hat{y}_1, \hat{y}_2, \dots, \hat{y}_K)$ is the prediction probability vector of the model for each class. This function measures the difference between the model's predicted and actual probability distributions by penalizing significantly higher incorrect predictions.

E. Model Evaluation

All CNN models that have been trained were evaluated to determine the best model to be implemented in the application. First, the evaluation was done by dividing the training data into two mutually exclusive subsets, namely data for training and data for validation, with a ratio of 80:20. The CNN model was then evaluated using classification accuracy on training data and validation data using Eq. (7),

$$Accuracy = \frac{t}{T} \times 100\% \quad (7)$$

where t is the number of face images in the dataset that are correctly classified and T is the number of all face images in the dataset.

From each scenario, one CNN model was selected with the highest accuracy on training and validation data. The three selected CNN models were reevaluated using testing data to find the best model. CNN models were evaluated on testing data by calculating classification accuracy using Eq. (7). Furthermore, because the number of face images in each class is not the same, CNN models were also evaluated in each class by calculating the precision, recall, and F_1 score values using Eqs. (8), (9), and (10), respectively,

$$precision = \frac{TP}{FP + TP} \quad (8)$$

$$recall = \frac{TP}{FN + TP} \quad (9)$$

$$F_1\ score = 2 \times \frac{precision \times recall}{precision + recall} \quad (10)$$

where *TP* is data in the positive class that is correctly classified, and *FN* is data in the positive class that is classified as a negative class. *FP* is data in the negative class that is classified as a positive class. Precision measures how accurate the model is in recognizing face images in the positive class. Recall measures how well the model can identify all positive class face images. F_1 score is the harmonic mean of precision and recall and combines the information from both metrics. F_1 score is useful for finding a balance between precision and recall, especially when the positive classes in the dataset are not balanced with the negative classes [30].

F. Implementation on The Application

The last step in this research is implementing the best CNN model obtained on the application to detect student engagement in online lectures using face images. The application was developed web-based so that users can access it without the need to install it on their respective computers. To detect student engagement, users needed to ask all students to open the camera on the online learning platform, then capture the face images of all students together. The captured face images were then uploaded to the application for detection, as shown in Figure 5. Furthermore, the uploaded image was displayed on the application page so that users could ensure that the uploaded image was as desired. If the image is as expected, the user must press the Predict Image button to run the student involvement detection process. The uploaded image was then processed and classified using the CNN model that had been trained. The prediction results of the CNN model were then displayed on the face image of each student as can be seen in Figure 6.

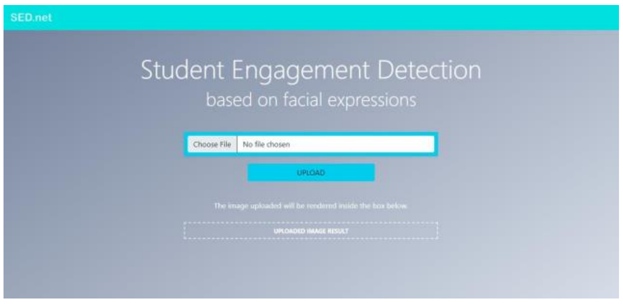
SED.net
Student Engagement Detection
based on facial expressions
Choose File
No file chosen
UPLOAD
The image uploaded will be rendered inside the box below.
UPLOADED IMAGE RESULT

Figure 5. The Page for Uploading Face Images.

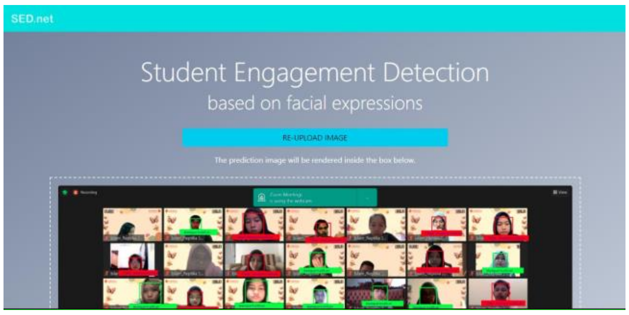
SED.net
Student Engagement Detection
based on facial expressions
RE-UPLOAD IMAGE
The prediction image will be rendered inside the box below.

Figure 6. Example of Detection Results.

PHP, a server-side scripting language, was used throughout the development of the web application. In addition, the step of image processing, as well as the development of the classification model, were both implemented using the Python programming language. OpenCV [31], Scikit-learn [32], and TensorFlow [33] were open-source libraries utilized for image processing, partitioning the dataset, and training the CNN model, respectively.

III. RESULTS AND DISCUSSION

The experimental results of detecting student engagement using 20 CNN models generated by the random search algorithm in three scenarios in testing and validation stages are tabulated in Table 2. As shown in Table 2, the performance of the CNN models varies for all scenarios. In scenario 1, the training and validation accuracy ranged from 75% to 92%. Some models, such as Model 3, Model 9, and Model 16, showed better performance with accuracies above 85% in both the training and validation stages. However, some other models, such as Model 11, Model 14, and Model 19, showed lower performance with accuracy below 80%. Model 9 achieved the highest training accuracy rate of 88% and the highest validation accuracy rate of 92%.

In scenario 2, some models, such as Model 5, Model 6, and Model 14, achieved high accuracy in both the training and validation stages, with validation accuracy reaching over 90%. However, some models, such as Model 7, Model 16, and Model 17, also showed lower accuracy below 80%, especially at the validation stage. In this scenario, Model 6 achieved the highest validation accuracy of 94%.

Table 2. The Performance of CNN Models in Training and Validation Data

Model	Accuracy (%)					
	S_1		S_2		S_3	
	T	V	T	V	T	V
Model 1	84	84	79	84	76	78
Model 2	80	80	79	80	84	86
Model 3	87	86	82	82	57	51
Model 4	83	86	80	90	80	80
Model 5	84	80	85	92	79	82
Model 6	78	80	81	**94**	86	**90**
Model 7	83	86	78	75	57	51
Model 8	82	82	80	78	57	51
Model 9	88	**92**	76	84	80	82
Model 10	81	80	82	80	81	80
Model 11	75	76	82	82	57	51
Model 12	78	84	80	82	57	51
Model 13	85	88	83	86	57	51
Model 14	75	76	87	90	57	51
Model 15	81	84	81	82	57	51
Model 16	87	90	76	76	78	78
Model 17	87	88	79	78	86	84
Model 18	81	82	82	82	81	88
Model 19	75	75	80	80	57	51
Model 20	79	82	79	80	57	51

S_1: Scenario 1, S_2: Scenario 2, S_3: Scenario 3
T: Training, V: Validation

In Scenario 3, Model 2, Model 6, and Model 18 showed relatively high accuracy in both training and validation. Model 6 achieved the highest accuracy of 94% at the validation stage. However, there were many models that showed low accuracy of below 60% in both training and validation. Also, in this scenario, most models had a high training accuracy, but their validation accuracy significantly dropped. This may indicate overfitting of the models. Among the three scenarios, the models generated in scenario 3 had the lowest performance compared to the previous two scenarios. Hyperparameter values of CNN models that achieved the best performance from all scenarios can be seen in Table 3.

Table 3. Hyperparameter Values for the Best CNN Models From All Scenarios

Layer	N	Hyper-parameter	Value		
			S_1	S_2	S_3
Convo-lutional	3	Filter's number	64, 288,	416, 256,	160, 160,

Layer	N	Hyper-parameter	Value		
			S_1	S_2	S_3
			160	256	288
		Kernel size	3 × 3	3 × 3	1 × 1
		Activation function	ReLU	ReLU	tanh
Max pooling	3	Kernel size	2 × 2	2 × 2	2 × 2
Fully con-nected	1	Node size	416	416	160
		Activation function	ReLU	ReLU	tanh
Output	1	Node size	2	2	2
		Activation function	Soft-max	Soft-max	Soft-max

N: layer size, S_1: Scenario 1, S_2: Scenario 2, S_3: Scenario 3

Furthermore, the three best CNN models from each scenario were re-evaluated using the testing data, and the results are tabulated in Table 4. The experimental results of student engagement detection in online lectures show consistent performance on the testing data. In the three scenarios, the accuracy of the developed CNN model reached 83%, 86%, and 81% in scenarios 1, 2, and 3, respectively. This shows that the model has a high level of accuracy in detecting student engagement, especially in scenario 2. In addition, the recall and precision values also experienced a comparable increase, with the highest recall and precision reaching 86% and 87%, respectively, in the best CNN model from scenario 2. This shows that the CNN model can effectively recognize students who are engaged in online lectures with a minimal error rate. The F1 score, which combines recall and precision, also shows a good level of balance in model performance, with the highest F1 score value reaching 86% in the CNN model from scenario 2. Overall, the results of this experiment show that the best CNN model from scenario 2 can be an effective tool in detecting student engagement in online lectures using facial images.

The method proposed in this study shows a marked improvement in accuracy for detecting student engagement compared to previous research. As can be seen in Table 5, the proposed method achieved an 86% accuracy, surpassing the SVM with facial shape features which had 80% [13], SVM with Gabor features at 72.9% [19], VGG-B at 72.38% [12], and VGGNet at 76.9% [22]. These findings emphasize the superior performance of the proposed method, demonstrating a notable advancement in the accuracy of engagement detection over earlier methods.

Table 4. The Performance of CNN Models in Testing Data

Scenario	Accuracy (%)	Recall (%)	Precision (%)	F_1 score (%)
1	83	83	84	83
2	**86**	**86**	**87**	**86**
3	81	81	82	81

Table 5. Comparison the Proposed Method with the Previous Research on Student Engagement Detection

Method	Accuracy (%)
SVM with facial shape features [13]	80
SVM with Gabor features [19]	72.9
VGG-B [12]	72.38
VGGNet [22]	76.9
This study	86

IV. CONCLUSION

In this research, an application was developed to detect student engagement in online lectures based on facial expression recognition. A CNN model was trained using a dataset of facial expression images acquired from 320 student volunteers. The dataset included images of students who were engaged and disengaged during online lectures. The CNN model was developed based on the LeNet architecture and underwent hyperparameter tuning to optimize its performance. The best CNN model architecture was determined through experiments using three hyperparameter tuning scenarios.

The results showed that the developed application achieved promising results in detecting student engagement in online lectures with the best accuracy of 86%. The findings of this research highlight the potential of facial expression recognition using CNN models for evaluating student engagement in online lectures. The developed application provides a practical tool for lecturers to assess student involvement during synchronous online lectures. Additionally, the application could be extended to include real-time feedback and interaction features to further enhance student engagement and learning outcomes in online lectures.

REFERENCES

[1] F. Amiti, “Synchronous and asynchronous E-learning,” *Eur. J. Open Educ. E-Learning Stud.*, vol. 5, no. 2, 2020.

[2] Z. Libasin, A. R. Azudin, N. A. Idris, M. S. A. Rahman, and N. Umar, “Comparison of students’ academic performance in mathematics course with synchronous and asynchronous online learning environments during COVID-19 crisis,” *Int. J. Acad. Res. Progress. Educ. Dev.*, vol. 10, no. 2, pp. 492–501, 2021.

[3] O. B. Adedoyin and E. Soykan, “Covid-19 pandemic and online learning: the challenges and opportunities,” *Interact. Learn. Environ.*, pp. 1–13, 2020.

[4] X. Xie, K. Siau, and F. F.-H. Nah, “COVID-19 pandemic–online education in the new normal and the next normal,” *J. Inf. Technol. case Appl. Res.*, vol. 22, no. 3, pp. 175–187, 2020.

[5] F. Martin, T. Sun, M. Turk, and A. D. Ritzhaupt, “A meta-analysis on the effects of synchronous online learning on cognitive and affective educational outcomes,” *Int. Rev. Res. Open Distrib. Learn.*, vol. 22, no. 3, pp. 205–242, 2021.

[6] F. Firman, A. P. Sari, and F. Firdaus, "Aktivitas mahasiswa dalam pembelajaran daring berbasis konferensi video: refleksi pembelajaran menggunakan Zoom dan Google Meet," *Indones. J. Educ. Sci.*, vol. 3, no. 2, pp. 130–137, 2021.

[7] B. R. A. Febrilia, I. C. Nissa, P. Pujilestari, and D. U. Setyawati, "Analisis Keterlibatan dan Respon Mahasiswa dalam Pembelajaran Daring Menggunakan Google Classroom di Masa Pandemi Covid-19," *FIBONACCI J. Pendidik. Mat. Dan Mat.*, vol. 6, no. 2, pp. 175–184, 2020.

[8] E. Di Lascio, S. Gashi, and S. Santini, "Unobtrusive assessment of students' emotional engagement during lectures using electrodermal activity sensors," *Proc. ACM Interactive, Mobile, Wearable Ubiquitous Technol.*, vol. 2, no. 3, pp. 1–21, 2018.

[9] H. Monkaresi, N. Bosch, R. A. Calvo, and S. K. D'Mello, "Automated detection of engagement using video-based estimation of facial expressions and heart rate," *IEEE Trans. Affect. Comput.*, vol. 8, no. 1, pp. 15–28, 2016.

[10] N. Alyuz *et al.*, "Semi-supervised model personalization for improved detection of learner's emotional engagement," in *Proceedings of the 18th ACM International Conference on Multimodal Interaction*, 2016, pp. 100–107.

[11] B. De Carolis, F. D'Errico, N. Macchiarulo, and G. Palestra, "'Engaged Faces': Measuring and Monitoring Student Engagement from Face and Gaze Behavior," in *IEEE/WIC/ACM International Conference on Web Intelligence-Companion Volume*, 2019, pp. 80–85.

[12] O. Mohamad Nezami, M. Dras, L. Hamey, D. Richards, S. Wan, and C. Paris, "Automatic recognition of student engagement using deep learning and facial expression," in *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, 2020, pp. 273–289.

[13] L. Chen, C. Zhou, and L. Shen, "Facial Expression Recognition Based on SVM in E-learning," *IERI Procedia*, vol. 2, pp. 781–787, 2012, doi: https://doi.org/10.1016/j.ieri.2012.06.171.

[14] D. A. Pisner and D. M. Schnyer, "Support vector machine," in *Machine learning*, Elsevier, 2020, pp. 101–121.

[15] A. Asthana, J. Saragih, M. Wagner, and R. Goecke, "Evaluating AAM fitting methods for facial expression recognition," in *2009 3rd International Conference on Affective Computing and Intelligent Interaction and Workshops*, 2009, pp. 1–8.

[16] J. G. Daugman, "Complete discrete 2-D Gabor transforms by neural networks for image analysis and compression," *IEEE Trans. Acoust.*, vol. 36, no. 7, pp. 1169–1179, 1988.

[17] M. Lyons, S. Akamatsu, M. Kamachi, and J. Gyoba, "Coding facial expressions with gabor wavelets," in *Proceedings Third IEEE international conference on automatic face and gesture recognition*, 1998, pp. 200–205.

[18] J. Whitehill, Z. Serpell, Y.-C. Lin, A. Foster, and J. R. Movellan, "The faces of engagement: Automatic recognition of student engagementfrom facial expressions," *IEEE Trans. Affect. Comput.*, vol. 5, no. 1, pp. 86–98, 2014.

[19] P. Viola and M. J. Jones, "Robust real-time face detection," *Int. J. Comput. Vis.*, vol. 57, no. 2, pp. 137–154, 2004.

[20] G. Littlewort *et al.*, "The computer expression recognition toolbox (CERT)," in *2011 IEEE International Conference on Automatic Face & Gesture Recognition (FG)*, 2011, pp. 298–305.

[21] J. Whitehill *et al.*, "Towards an optimal affect-sensitive instructional system of cognitive skills," in *CVPR 2011 WORKSHOPS*, 2011, pp. 20–25.

[22] C. Pabba and P. Kumar, "An intelligent system for monitoring students' engagement in large classroom teaching through facial expression recognition," *Expert Syst.*, vol. 39, no. 1, p. e12839, 2022.

[23] S. Nurdiati, M. K. Najib, F. Bukhari, M. R. Ardhana, S. Rahmah, and T. P. Blante, "Perbandingan AlexNet dan VGG untuk Pengenalan Ekspresi Wajah pada Dataset Kelas Komputasi Lanjut," *Techno. Com*, vol. 21, no. 3, pp. 500–510, 2022.

[24] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv Prepr. arXiv1409.1556*, 2014.

[25] I. J. Goodfellow *et al.*, "Challenges in Representation Learning: A Report on Three Machine Learning Contests BT - Neural Information Processing," 2013, pp. 117–124.

[26] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Commun. ACM*, vol. 60, no. 6, pp. 84–90, 2017.

[27] R. C. Gonzalez, *Digital Image Processing*, 4th ed. Prentice Hall, 2017.

[28] J. Bergstra and Y. Bengio, "Random search for hyper-parameter optimization," *J. Mach. Learn. Res.*, vol. 13, no. 1, pp. 281–305, 2012.

[29] P. K. Diederik and J. L. Ba, "Adam: a Method for Stochastic Optimization International Conference On Learning Representations, 2015." 2015.

[30] E. Alpaydin, *Introduction to machine learning*, 2nd ed. Cambridge, Massachusetts: MIT press, 2014.

[31] G. Bradski and A. Kaehler, *Learning OpenCV: Computer Vision with the OpenCV Library*. Sebastopol, CA: O'Reilly Media, Inc., 2008.

[32] F. Pedregosa *et al.*, "Scikit-learn: Machine learning in Python," *J. Mach. Learn. Res.*, vol. 12, pp. 2825–2830, Oct. 2011.

[33] M. Abadi *et al.*, "TensorFlow: A System for Large-Scale Machine Learning," in *12th USENIX symposium on operating systems design and implementation (OSDI 16)*, 2016, pp. 265–283.