education sciences

MDPI

Article

Educational Innovation Faced with COVID-19: Deep Learning for Online Exam Cheating Detection

Intan Nurma Yulita [1,*], Fauzan Akmal Hariz [2], Ino Suryana [2] and Anton Satria Prabuwono [3]

[1] Research Center for Artificial Intelligence and Big Data, Universitas Padjadjaran, Bandung 40132, Indonesia
[2] Department of Computer Science, Faculty of Mathematics and Natural Sciences, Universitas Padjadjaran, Sumedang 45363, Indonesia
[3] Faculty of Computing and Information Technology in Rabigh, King Abdulaziz University, Rabigh 21911, Saudi Arabia
* Correspondence: intan.nurma@unpad.ac.id

Abstract: Because the COVID-19 epidemic has limited human activities, it has touched almost every sector. Education is one of the most affected areas. To prevent physical touch between students, schools and campuses must adapt their complete learning system to an online environment. The difficulty with this technique arises when the teachers or lecturers administer exams. It is difficult to oversee pupils one by one online. This research proposes the development of a computer program to aid in this effort. By applying deep learning models, this program can detect a person's activities during an online exam based on a web camera. The reliability of this system is 84.52% based on the parameter F1-score. This study built an Indonesian-language web-based application. Teachers and lecturers in Indonesia can use this tool to evaluate whether students are cheating on online exams. Unquestionably, this application is a tool that may be utilized to develop distance learning educational technology in Indonesia.

Keywords: COVID-19; deep learning; web-based application; online exams

check for updates

Citation: Yulita, I.N.; Hariz, F.A.; Suryana, I.; Prabuwono, A.S. Educational Innovation Faced with COVID-19: Deep Learning for Online Exam Cheating Detection. *Educ. Sci.* **2023**, *13*, 194. https://doi.org/10.3390/educsci13020194

Academic Editors: Maria Meletiou-Mavrotheris, Konstantinos Katzis, Angelos Sofianidis, Nayia Stylianidou and Panagiota Konstantinou-Katzi

Received: 28 December 2022
Revised: 5 February 2023
Accepted: 9 February 2023
Published: 12 February 2023

1. Introduction

Many activities have changed due to the COVID-19 pandemic. Various countries have attempted to overcome this, including Indonesia, which has imposed large-scale social restrictions [1]. The policy affects many activities in various fields, one of which is education. In the field of education, teaching methods that previously used face-to-face methods have shifted to online teaching methods [2]. In the teaching and learning process, a method is needed to assess the achievements of students in the form of an exam. However, the implementation of this exam still cannot be separated from its relation to academic cheating [3,4]. Cheating activity has been a problem in the world of education for a long time; coupled with the pandemic conditions have increasingly made it worse [5–7]. To detect cheating activity, a strict and organized control system is required.

Human activity recognition (HAR) is one of the issues that is receiving a lot of attention as technology advances at a rapid pace [8–10]. Because of its widespread usage, HAR has been the subject of extensive investigation. It may be utilized in a variety of situations, from monitoring ill individuals to monitoring people in houses that employ the smart home idea [11]. It can aid in decision-making or produce early warnings. HAR can also be implemented to detect cheating. To do so, the system must develop a model that can distinguish human behavior based on recorded camera captures generated by someone taking an online exam.

Deep learning can be used for detection in this investigation. It is a machine learning methodology that uses many layers of nonlinear information processing to achieve feature extraction, pattern identification, and classification [12,13]. It is a frequently utilized approach for HAR [14–16]. Its architecture makes use of artificial neural networks. It is so-called because of its numerous connected layers and neurons. When new data is received, the algorithm is taught to recognize patterns and categorize various sorts of information [17]. The primary distinction between machine learning and deep learning is that feature extraction in machine learning must be manually constructed, which requires a significant amount of time and work. Meanwhile, deep learning extracts characteristics for categorization automatically. On the other hand, a significant quantity of data is needed to train the algorithm [18].

Convolution neural network (CNN) is the deep learning algorithm used in this work. It is believed to be the best model for handling object detection and recognition issues and may be used to solve difficulties in data classification with a high degree of accuracy [14]. It is a deep learning method that is meant to interpret two-dimensional input, such as images or speech [19]. It was motivated by how people process and develop a visual perception in order to distinguish or detect an item in a digital picture. It uses a supervised learning approach to categorize labeled data. It is frequently used to distinguish objects or perspectives, as well as to detect, segment, and classify images [20]. It is a technology that may also be used to detect human activities [21]. It is created by stacking the three variants of the layer: convolutional, pooling, and fully connected layers [22].

There are several CNN architectures. MobileNet is one of them. The design differs from other CNN architectures in that the thickness of the filter on the convolution layer matches the thickness of the input. The convolution is divided into depth-wise and pointwise convolution [23]. There are inputs and outputs between models at the bottleneck, while the inner layers embody the model's capacity to adapt inputs from lower-level concepts. As a result, shortcuts between bottlenecks enable faster training and higher accuracy [24]. MobileNetV2 is a CNN architecture for image detection that evolved from MobileNetV1 [25]. It employs deep and directed convolution. With fewer parameters, it achieves higher accuracy than MobileNetV1. It introduces two new capabilities: linear bottlenecks and shortcut connections between bottlenecks [26].

The thickness of the filter on the convolution layer based on the input which distinguishes the MobileNet design from other CNN architectures. It separates convolution into depth-wise and point-wise convolution, allowing for quicker and more accurate training [24]. This study creates an Indonesian-language web-based application. The application's input is a video from participant recordings during online tests. According to the researchers' observations, no program exists that evaluates recordings from online tests asynchronous. Another use under development is synchronous detection analysis, which necessitates a huge internet limit. Indonesia is an archipelago with varying levels of internet availability across the country. If they share the internet network quota, the adoption of a synchronous cheat detection program during the exam's implementation would be disturbed. As a result, this study suggests an asynchronous use. Furthermore, the program employs Indonesian as the language of instruction to make it easier for teachers and lecturers in Indonesia to use. Every instruction in this program is designed to be easier to grasp for all Indonesians due to the usage of the national language of Indonesia. This research proposes an asynchronous, website-based application in Indonesian that can be used to identify online exam cheating. This program can be utilized by individuals from diverse educational levels and backgrounds. Only if there is a web camera in front of the computer monitor pointing directly at the examinee's face may monitoring be performed.

2. Related Works

Academic dishonesty is a complicated problem that is generally understood as occurring when students engage in plagiarism, copying and pasting, looking at other people's work, and data fabrication [27]. Academic dishonesty undermines the integrity of educational institutions by resulting in inaccurate academic evaluations and possibly inaccurate grading of students. Academic integrity is seriously threatened by cheating on academic tasks, which is a serious ethical issue. Academic dishonesty has an enormous impact on the credibility of educational institutions and the trustworthiness of their students. To detect dishonesty, several methods have been used. Institutions have implemented a variety of methods to prevent cheating, from technology-based solutions to moral education [28]. By using technology, such as digital scanning of essays, turnitin.com, or software to detect plagiarism, educational institutions can ensure that their students are held accountable for their own work [29]. The advancement of technology has accelerated research in this field.

Alrubaish et al. suggest an improvement to current algorithms in order to develop a model that identifies cheating intent [30]. The algorithm is backed by a collection of technologies and equipment, including a heat detector connected to a security camera and augmented by an eye-tracking system. When students want to cheat, their bodies release a particular range of heat owing to the relationship between their bodies and their emotions. The radiated heat will cause the camera to concentrate and detect the students' faces, after which it will detect their eyes and begin analyzing their movement to determine whether a student intends to cheat. Eventually, adopting this model would be highly beneficial in detecting the cheating intentions of students, and its application would not be confined to educational contexts alone; it could be implemented in other sectors with slight modification.

Using a 360-degree security camera, Turani et al. present a novel method for exam proctoring [31]. Mainly, online examinations' security is a key worry. Consequently, a delivery technology must not only guarantee the identification of a test-taker, but also the test's entire validity. This research investigates the use of the 360-degree security camera over the typical webcam in order to increase test security and reduce stressful limits. In order to avoid cheating, this research also proposes an automated proctoring paradigm that eliminates the need for real-time proctoring and any scheduling restrictions, utilizing machine learning methods to enhance the suggested system. To enable authentication and the seamless operation of the online test, a secure framework employing biometrics is utilized.

Another approach related to hardware was carried out by Atoum et al. The system hardware consists of a camera, a wear-cam, and a microphone to monitor the visual and aural surroundings of the testing site [32]. Their research describes a multimedia analytics system that administers online examinations automatically. The system consists of six fundamental components that continually assess the most important behavioral cues: user verification, text detection, voice detection, active window detection, gaze estimation, and phone detection. By merging the continuous estimating components and adding a temporal sliding window, we created higher-level features to classify whether a test-taker is cheating at any time throughout the exam.

Research developments rely not only on hardware, but also on the use of computer vision through a machine-learning approach. By taking advantage of advances in machine learning, computer vision has become increasingly efficient in processing images, object recognition, and tracking, enabling research to be more accurate and reliable. Exam cheating is frequently regarded as an abnormal occurrence. Strange postures or motions help researchers identify it [33–40]. This detection is accomplished effectively through the use of computer vision. Using machine learning, systems become more intelligent. With this increased intelligence, computer vision systems are able to detect suspicious behaviors more effectively than before. Technological advances in the field of processing using deep learning also have a direct impact. A number of studies have reported the use of this method for the detection of suspicious activity in both online and offline exams. Most of the approaches employed are CNN [41–44]. CNN is expanding its use of transfer learning for object identification in many applications, including pedestrian surveillance [45], industrial automation [46], and medical applications [47,48]. As presented in this study, it is hypothesized that transfer learning will likewise yield positive results for detecting online exam fraud. There are several programs for detecting online exam cheating; however, they are now only available in English, such as https://www.autoproctor.com, (accessed on 23 November 2022). Consequently, Indonesian-language software are still required to assist educators of all educational levels and backgrounds.

3. Methodology

The research carried out included collecting data on online exam activities in the form of video recordings extracted into frames, data preprocessing, in which data augmentation was carried out to enrich existing data, data splitting, which divided data into training data and test data, model creation and training, and model evaluation and application. It is shown in Figure 1.

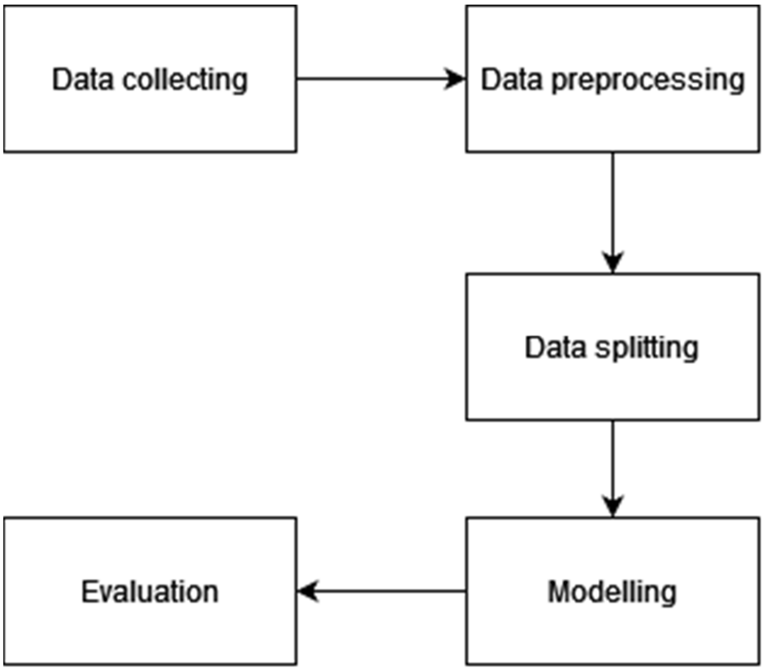


Figure 1. The proposed framework.

3.1. Data Collecting

The video data recorded from webcams of several persons taking online tests were utilized in this investigation. The dataset was obtained from Michigan State University's Computer Vision Lab and is available at the following address: http://cvlab.cse.msu.edu/project-OEP.html (accessed on 18 February 2022). There were 24 participants in the dataset who took the online exam. The data was separated into 15 participants that played students and 9 participants that were actual students taking online exams. Each participant had a video file that was a recording of online test activities, and there was a ground truth of the actions that occur in the video. This study's online exam activities included *No Cheat*, *Read Text*, *Ask Friend*, and *Call Friend*. Each subject was extracted into a frame and classified based on ground truth. The image data frame was then tagged with the appropriate class.

3.2. Data Preprocessing

Data preprocessing is the process of equalizing data sizes and augmenting data for all data [36]. Its role is to alter the data so that it fits the intended format, resulting in more and more diverse data and a more accurate model. In this work, data augmentation employed a transformation to generate numerous copies of frames with varying variances. Rotating, shifting, flipping, and zooming were some of the augmentation techniques used. This strategy did not alter the target class, but it gave a new viewpoint on catching objects in new frames [37]. To eliminate the class imbalance in the data, the data augmentation approach was also employed as an oversampling method. It not only expanded the data set but also introduced some variety, allowing the model to generate better classifiable predictions on previously unknown data. Furthermore, when trained on new, slightly modified frames, the models became more resilient. Some of the augmentation strategies used in this study are listed below.

Random rotation: This is an augmentation approach that allows the model to be object orientation invariant [38]. Data augmentation in the form of rotation range allows for random rotation of frames across any degree between 0 and 360. Some pixels migrate outside the picture when the frame rotates, leaving empty spaces that could be filled.

Random shifts: This is a method used to compensate for the fact that objects are not constantly in the center of the frame. This issue can be resolved by moving the frame pixels horizontally or vertically. The height shift range is used for vertical frame shifts, whereas the width shift range is utilized for horizontal shifts.

Random Flip: One of the augmentation strategies used to flip frames is random flip [39]. The horizontal flip is used to flip the frame horizontally, and the vertical flip is used to flip the frame vertically. However, in order for the model to generate classification results based on the class, the augmentation strategy utilizing random flip must have a frame that is symmetrical with the item in the original frame.

Random Zoom: This is an augmentation method that uses a zoom range to randomly increase or shrink the picture size.

3.3. Data Splitting

The process of separating data into categories is known as data splitting. At this point, the acquired data frames were separated into training and test data. The training data was derived from the subject frames, which were made up of 14 subject actors that play students, totaling 9000 frames. The test data was collected from subject frames of 9 original student subjects and an actor subject, totaling 7000 frames. The training data was then separated into training data and validation data. In this work, the validation technique was 5-stratified k-fold cross-validation. The data was divided into three categories: training, validation, and test data. The training data was used to train the model, and the validation was used to test the model that was developed in each iteration of the hyperparameter testing process. The test results were used to assess the performance of the final model.

3.4. Modelling

This study created a model utilizing the CNN technique with the MobileNetV2 architecture. It contains fewer parameters than previous designs [41]. Figure 2 depicts the simplified architecture in this study, which was updated in the head model section. Here are some descriptions of the phases completed:

1. The input was a 224 by 224-pixel data frame.
2. The data was run through a 2D convolution with dimensions of 224 × 224 × 3, where 3 was the RGB color model channel in the frame. The convolution layer was used to build a feature map that would be blended from the preceding layers.
3. Data passed through the main components in MobileNetV2, namely the bottleneck residual blocks, which could number up to 17 and each had a different number of repetitions, pixel sizes, and dimensions, as well as the bottleneck residual blocks that data passed through, namely between stride 1 block and stride 2 blocks. The first block had dimensions of 112 × 112 × 32, the second block had dimensions of 112 × 112 × 16, the third block had dimensions of 56 × 56 × 24, the fourth block had dimensions of 28 × 28 × 32, the fifth block had dimensions of 14 × 14 × 64, the sixth block had dimensions of 14 × 14 × 96, and the last block had dimensions of 7 × 7 × 160. The bottleneck residual block is an important component of the MobileNetV2 design. The bottleneck residual block had two strides, one with one block and one with two blocks. There were various steps in a stride with one block, as follows:
    - Inputs.
    - Convolution using a 1 × 1 kernel size and the ReLU6 activation function.
    - Depthwise Convolution using a 3 × 3 kernel size and the ReLU6 activation function.
    - Convolution with a 1 × 1 kernel.

- Add, which is also known as the shortcut connection, was the addition of input from the previous process to the output of the current process, which was symbolized by an arrow pointing from input to add. The only distinction between the two blocks is the absence of an add stage. Essentially, the more layers utilized in the model, the higher the accuracy score.

4. The data was processed using a 2D convolution with a kernel size of 1 × 1 and dimensions of 7 × 7 × 320.
5. The data was routed through the head model, which was a model added to the MobileNetV2 architecture's head section.
    - The first layer in the head model was a 2D convolution with a kernel size of 3 × 3 and dimensions of 7 × 7 × 1280.
    - The data was sent via 2D max pooling with a pool size of 2 × 2, which yielded the maximum value of the region of the frame covered by the kernel to minimize the size of the feature map with dimensions of 5 × 5 × 100.
    - The data passed through a flattening with a dimension of 400, which converted data from the previous layer with multiple dimensions into one dimension so that it was utilized in the following layer, namely the fully linked layer.
    - The data passed through a thick layer that got a feature map from the previous layer to link all data from one layer to each activation unit of the next layer, which was utilized for classification. The dense layer portion contained counts and units, where the count was the number of dense layers utilized and units was the number of units in the dense layer.

In the last layer, data passed through the last dense layer, which was utilized to generate activity probabilities in data frames with four classifications depending on the dataset used. There are various criteria that must be taken into account while deciding the number of layers. When the system uses a few layers, it may minimize the performance value and even enter an underfitting situation, but when it uses an excess layer, it can enter an overfitting state. Batch size is the number of distributed data samples to a neural network. The system cannot give the complete dataset to the neural network at once; therefore, it must divide the dataset into multiple sections. The ideal batch size must be determined for the system to function optimally. As a result, an experiment was required to establish the number of layers and batch size employed in this study. The model training procedure was carried out using training data in this work by evaluating the hyperparameters that were set with a maximum epoch of 20 and employing early stopping to prevent model overfitting. Furthermore, the best batch size and number of dense layers were sought in this work. Table 1 shows the hyperparameter values from this work.

Table 1. Hyperparameter values.

Hyperparameter	Value
Optimizer	Adam
Activation	Softmax
Max epoch	20
Early Stopping	10
Learning Rate	0,0001
Batch Size	[16; 32; 64]
Dense Layer	[1; 3; 5; 7]

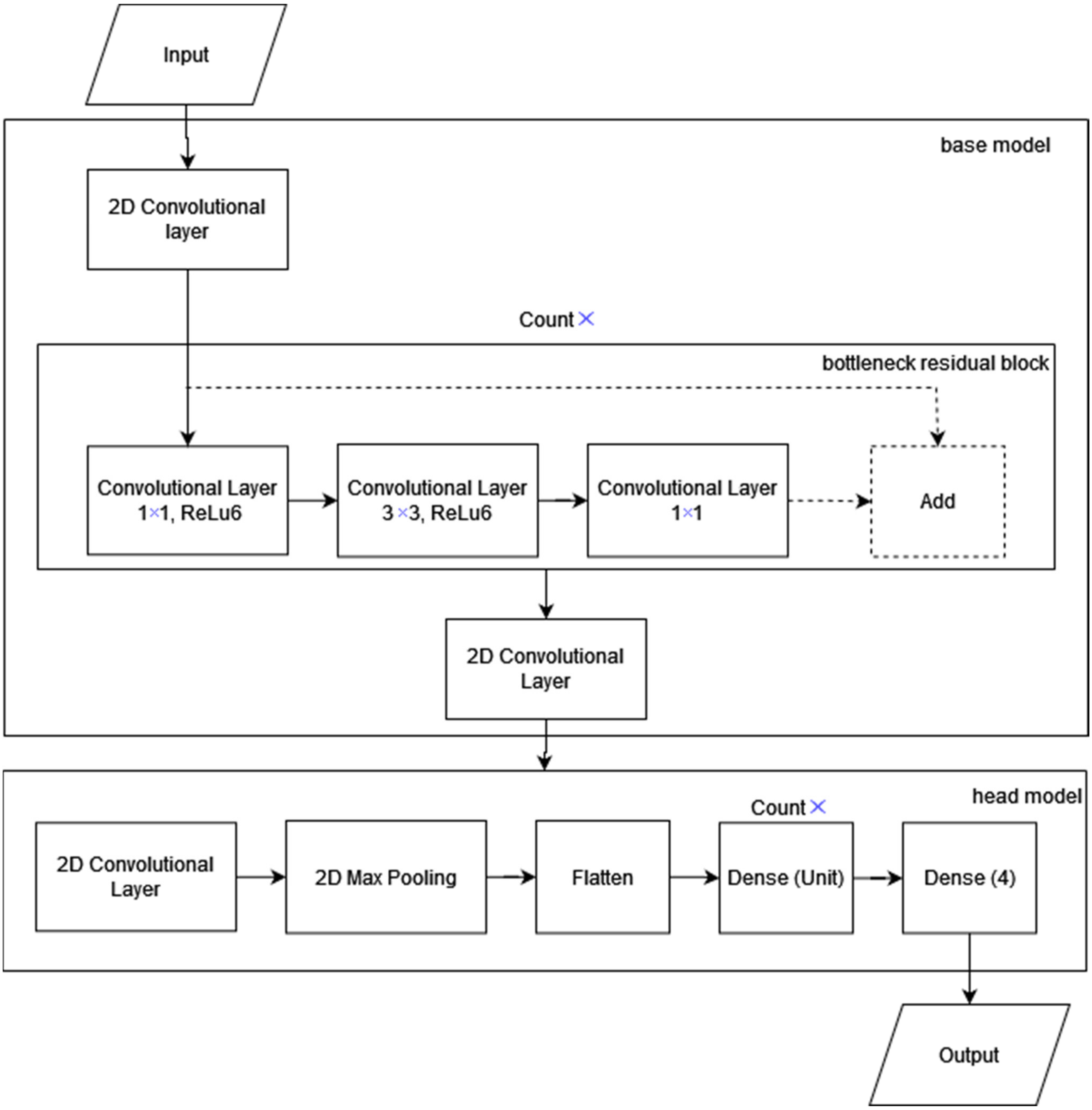


Figure 2. Simplified MobileNetV2 Model Architecture.

3.5. Evaluation

The results of the training can be used to determine if the machine can learn properly or not by using test data. These are examined in order to ensure that the model operates properly. The value of the confusion matrix is used to evaluate the model. A confusion matrix, also known as an error matrix, is a table used to evaluate machine learning models. This table is designed to better understand how well a model or classifier does in providing the correct response, meaning predicting the correct class of data for the correct class. Several alternative model assessment scores, such as accuracy, recall, and F1-score, may be generated based on the combination of values in the confusion matrix table [42].

- Precision: Precision, also known as a positive predictive value, is the ratio of samples properly predicted as positive to all samples projected to be positive.

- Recall: The percentage of samples that are accurately predicted as positive to all positive samples is known as recall, also known as sensitivity or true positive rate.
- F1-score: The harmonic mean or ratio of the average comparison of the weighted precision and recall values is the F1-score. It considers precision and recall values simultaneously, allowing it to quantify model performance more precisely for imbalanced class scenarios. As a result, the F1-score was used in this study to assess the model's performance.

4. Results and Discussion

Hyperparameters are one of the aspects that influence model performance in deep learning. The optimal hyperparameters will result in a model that performs optimally. The model's performance was evaluated using the average F1-score derived from the five-fold cross-validation results in this study.

4.1. Analysis of Batch Size

The testing was performed in this study with batch sizes 16, 32, and 64. Table 2 shows the results of the batch size test. According to the findings of these experiments, a batch size of 16 yields the greatest F1-score. It demonstrates that lowering batch size improved performance. This was seen by the lower F1-scores obtained with batch sizes of 64 and 32, as opposed to the greatest F1-score obtained with a batch size of 16. According to the batch size test findings, choosing a batch size number that was not too large was adequate for the study being conducted. A big batch size speeded up model convergence but made it difficult for the model to learn patterns from the data provided since there were so many of them. The ideal F1-score value while employing a batch size of 16 was determined in this investigation.

Table 2. Test results.

Number of Dense Layer	Batch Size	Averaged F1-Score (%)
1	64	66.53
1	32	73.15
1	16	73.65
3	64	69.02
3	32	75.43
3	16	76.73
5	64	70.80
5	32	71.58
5	16	80.81
7	64	71.87
7	32	71.03
7	16	77.55

4.2. Analysis of the Number of Dense Layers

The number of dense layers was the hyperparameter study done for the model. This study evaluated dense layers of one, three, five, and seven layers stacked on top of each other. Table 2 also shows the results of testing the number of dense layers. It demonstrates that increasing the number of dense layers initially improved model performance by raising the F1-score value from the dense layer by one to three. However, while utilizing a batch size of 32 and adding as many as five dense layers, the model performance deteriorated. The batch size was 16, and the dense layer was five, indicating that the model performed perfectly. The addition of a dense layer to as many as seven reduced the F1-score value. The dense layer test results showed that adding a dense layer might increase the model's ability to understand data, but exceeding a certain number of dense layers led the model to progressively overfit and offer less-than-ideal performance. Increasing the number of dense layers could improve the model's learning ability. However, as the model's learning ability increased, it extracted more complex patterns from the data presented. In general, this was advantageous since the model learned more to better predict the data. However, it also ran the danger of overfitting the created model. Models could predict well-studied data but did poorly when predicting data that had never been seen before. As a result, while utilizing a dense layer of five, this study had an ideal F1-score value.

4.3. Final Evaluation of Model

After multiple testing with five dense layers and a batch size of 16, the inquiry yielded perfect hyper-parameter findings. The model was then evaluated again with the previously separated test data. The model was evaluated based on the results of each fold in cross-validation and each class. Table 3 displays the results of the five-fold stratified cross-validation generated from the model with the best average prediction outcomes. The model on the fifth fold performed the best.

Table 3. 5-fold cross-validation.

Fold	Averaged F1-Score (%)
1	79.82
2	73.63
3	84.31
4	81.79
5	84.52
Macro Average	80.81

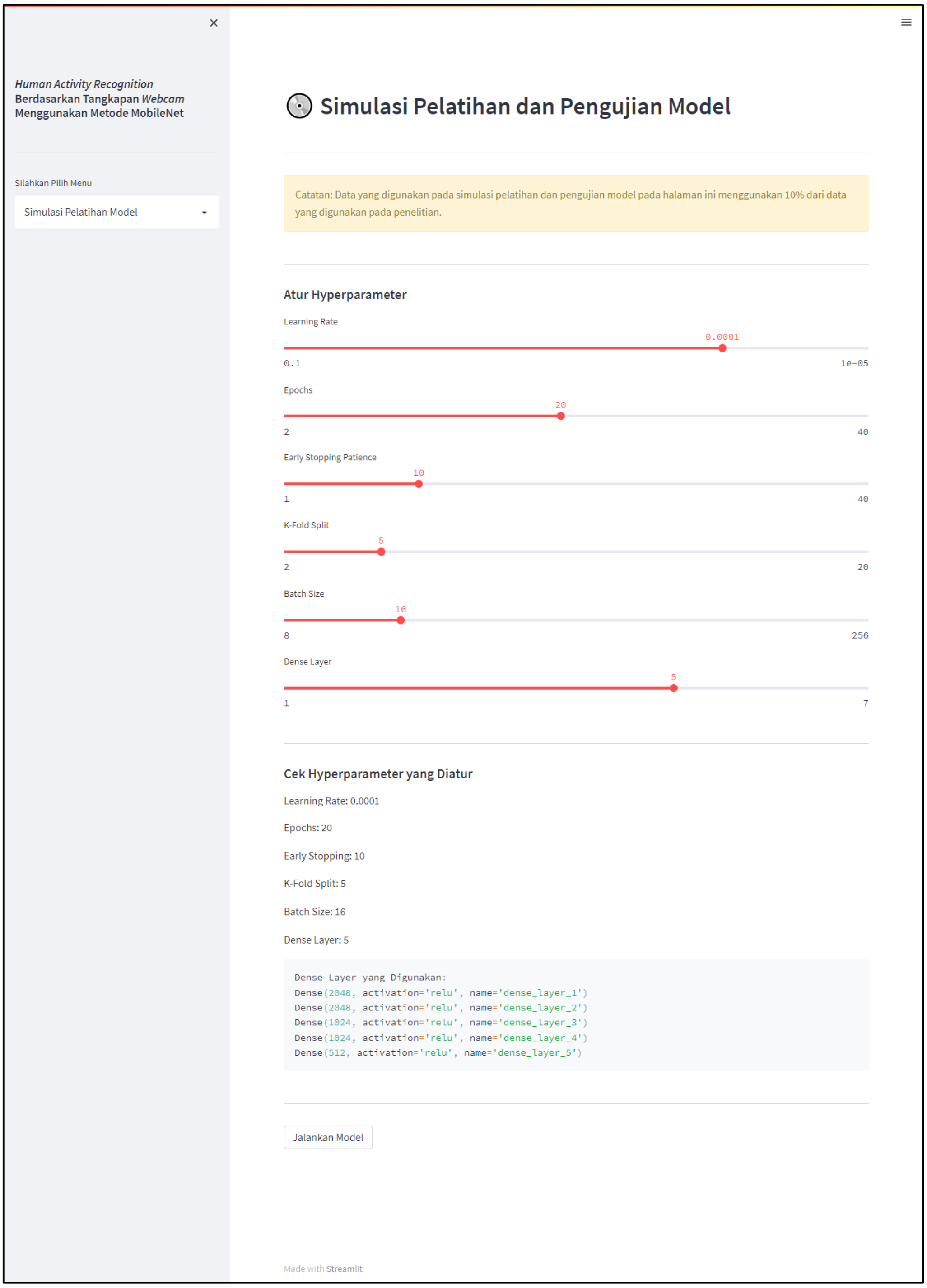
Table 4 shows the accuracy, recall, and F1-score values for each class on the fifth fold. Class 2 (*Ask Friend*) received the lowest F1-score of 80.37%, while class 1 (*Read Text*) received the highest of 93.19%. This was due to an imbalanced class in which the class with the greatest data dominated the sample space. The overall performance of the developed model received an F1-score of 84.52%.

Table 4. Evaluation results per class.

Class	Precision (%)	Recall (%)	F1-Score (%)
0 (No Cheat)	93.38	71.20	80.79
1 (Read Text)	90.06	96.55	93.19
2 (Ask Friend)	68.64	96.93	80.37
5 (Call Friend)	92.18	76.70	83.73
Macro Average	86.07	85.34	84.52

4.4. Application Implementation

This study's implementation takes the shape of a web-based application. This application also provides a menu for the MobileNet architecture's model training procedure with newly entered user data. There is no maximum dataset size for training the model in this application. This is the benefit of employing the transfer learning technique from deep learning, as is the case with the MobileNet architecture utilized in this study. A huge dataset was used to pre-train the model. Fine tuning during training consists merely of updating and adjusting the parameters for the newly trained dataset. The larger the dataset, the more accurate the resulting model. Nonetheless, as a result, the required hardware capacity must be raised. The in-training option can be disregarded because, by default, a model has already been trained. The training menu is depicted in Figure 3.



Human Activity Recognition Berdasarkan Tangkapan *Webcam* Menggunakan Metode MobileNet

Silahkan Pilih Menu

Simulasi Pelatihan Model

Simulasi Pelatihan dan Pengujian Model

Catatan: Data yang digunakan pada simulasi pelatihan dan pengujian model pada halaman ini menggunakan 10% dari data yang digunakan pada penelitian.

Atur Hyperparameter

Learning Rate: 0.0001 (0.1 – 1e-05)

Epochs: 20 (2 – 40)

Early Stopping Patience: 10 (1 – 40)

K-Fold Split: 5 (2 – 20)

Batch Size: 16 (8 – 256)

Dense Layer: 5 (1 – 7)

Cek Hyperparameter yang Diatur

Learning Rate: 0.0001

Epochs: 20

Early Stopping: 10

K-Fold Split: 5

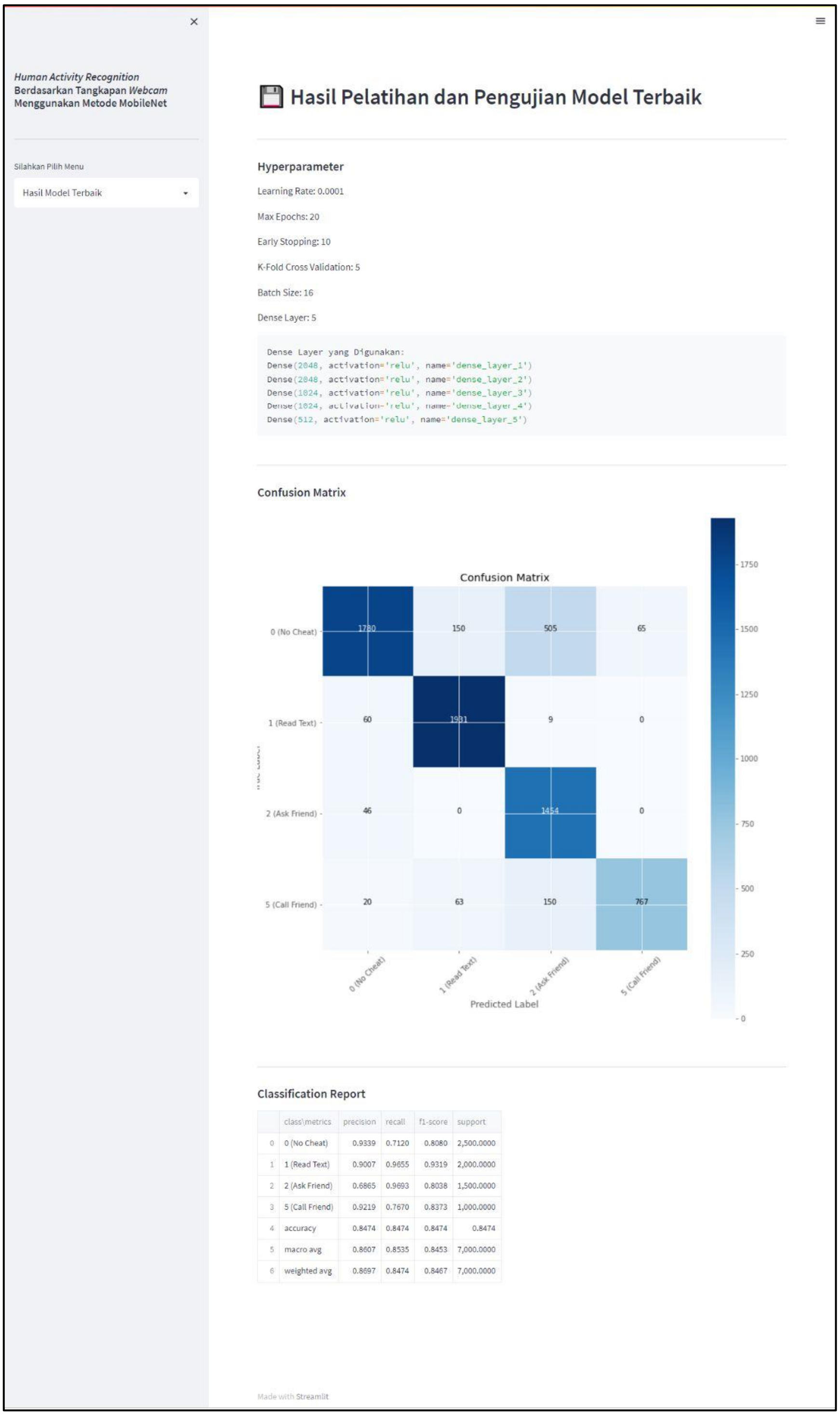
Batch Size: 16

Dense Layer: 5

```
Dense Layer yang Digunakan:
Dense(2048, activation='relu', name='dense_layer_1')
Dense(2048, activation='relu', name='dense_layer_2')
Dense(1024, activation='relu', name='dense_layer_3')
Dense(1024, activation='relu', name='dense_layer_4')
Dense(512, activation='relu', name='dense_layer_5')
```

Figure 3. Training page. The application is presented in Indonesian. The user can set hyperparameter values in the middle of the page. The bottom section presents the hyperparameter values that have been set by the user. The bottom button performs training from the deep learning model according to the given hyperparameter value when pressed.

If the user selects in-app retraining, the app also displays an evaluation of the in-app training. This program gives a menu for evaluation as depicted in Figure 4. This page can assist users in deciding whether to select a trained model. The presentation of confusion matrix information will undoubtedly aid in this determination procedure.



Human Activity Recognition Berdasarkan Tangkapan *Webcam* Menggunakan Metode MobileNet

Silahkan Pilih Menu: Hasil Model Terbaik

Hasil Pelatihan dan Pengujian Model Terbaik

Hyperparameter

Learning Rate: 0.0001

Max Epochs: 20

Early Stopping: 10

K-Fold Cross Validation: 5

Batch Size: 16

Dense Layer: 5

```
Dense Layer yang Digunakan:
Dense(2048, activation='relu', name='dense_layer_1')
Dense(2048, activation='relu', name='dense_layer_2')
Dense(1024, activation='relu', name='dense_layer_3')
Dense(1024, activation='relu', name='dense_layer_4')
Dense(512, activation='relu', name='dense_layer_5')
```

Confusion Matrix

Confusion Matrix

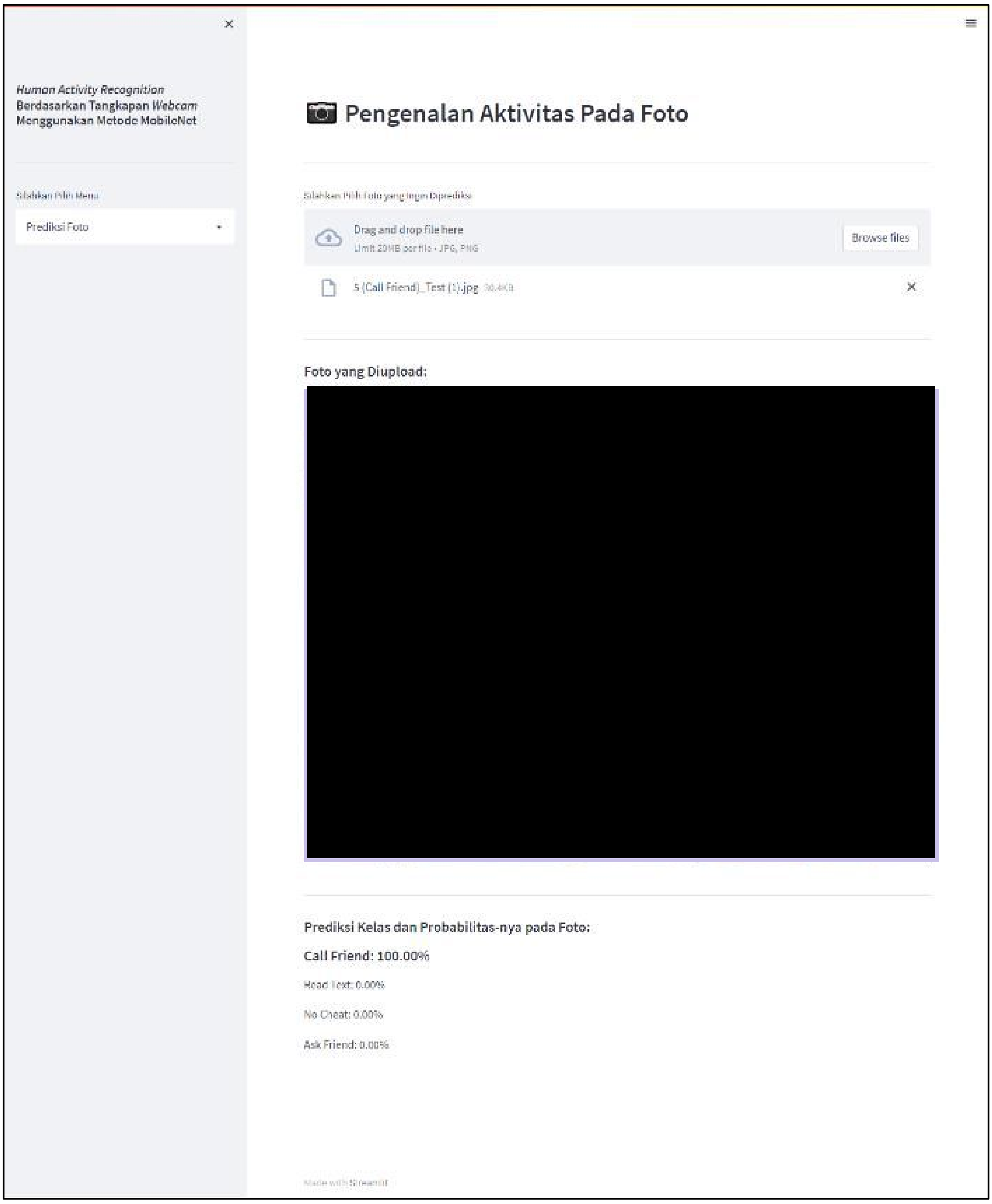
	0 (No Cheat)	1 (Read Text)	2 (Ask Friend)	5 (Call Friend)
0 (No Cheat)	1780	150	505	65
1 (Read Text)	60	1931	9	0
2 (Ask Friend)	46	0	1454	0
5 (Call Friend)	20	63	150	767

Classification Report

	class\metrics	precision	recall	f1-score	support
0	0 (No Cheat)	0.9339	0.7120	0.8080	2,500.0000
1	1 (Read Text)	0.9007	0.9655	0.9319	2,000.0000
2	2 (Ask Friend)	0.6865	0.9693	0.8038	1,500.0000
3	5 (Call Friend)	0.9219	0.7670	0.8373	1,000.0000
4	accuracy	0.8474	0.8474	0.8474	0.8474
5	macro avg	0.8607	0.8535	0.8453	7,000.0000
6	weighted avg	0.8697	0.8474	0.8467	7,000.0000

Figure 4. Evaluation page. This Indonesian language page provides the results of the training process. The top section contains the hyperparameter values used. The middle is the confusion matrix. The bottom section is a report on the results of the classification.

Figure 5 depicts the application's primary navigation menu. This program has two distinct user types. The first user is the one who wants to attempt retraining, and the second user is the one who wants to use it in practice. This second user can directly access the application by selecting the testing procedure from the application's main menu. Users can add images and videos relating to ongoing online exam activity. The application will generate class predictions directly from an image or video. Figure 5 depicts a menu that appears if the user uploads an image. As a result, the program displays the image as having a 100% Call Friend status, revealing proof of the examinee's dishonesty. If the user uploads a video, the system responds with two options: The first reaction is a summary of all the events in the video. This program shows the proportions of four classes. Apart from displaying a summary, the program also offers real-time class predictions while the movie is playing. This application makes it easier for instructors or lecturers to identify their students' activity during the exam.



Human Activity Recognition Berdasarkan Tangkapan *Webcam* Menggunakan Metode MobileNet

Silahkan Pilih Menu

Prediksi Foto

Pengenalan Aktivitas Pada Foto

Silahkan Pilih Foto yang Ingin Diprediksi

Drag and drop file here

Limit 20MB per file • JPG, PNG

5 (Call Friend)_Test (1).jpg 30.4KB

Foto yang Diupload:

[redacted]

Prediksi Kelas dan Probabilitas-nya pada Foto:

Call Friend: 100.00%

Read Text: 0.00%

No Cheat: 0.00%

Ask Friend: 0.00%

Made with Streamlit

Figure 5. Activity recognition when an image is uploaded to the application. The menu for uploading images is at the top. The middle area shows an image preview of the uploaded image. The bottom section has prediction for each class. The class with the highest likelihood is highlighted in bold and larger font.

5. Conclusions

This study uses HAR that employs a deep learning model based on the MobileNetV2 architecture. The data was derived from a video clip of a person taking an online exam using webcam capture. The dataset has four classes: 0 (*No Cheat*), 1 (*Read Text*), 2 (*Ask Friend*), and 3 (*Call Friend*). When employing hyperparameters such as a max epoch of 20, a learning rate of 0.0001, a batch size of 16, and a dense layer of five, the deep learning model with the MobileNetV2 architecture achieved optimal performance. Overall, the evaluation findings have an F1-score of 84.52%. The primary objective of this research, after the optimal model was identified, is the creation of an Indonesian-language web-based application. Teachers and lecturers in Indonesia can use this program to determine if students are cheating on online tests with the help of this application. This program is unquestionably a tool that may be used to advance educational technology in Indonesia.

Author Contributions: Conceptualization, I.N.Y.; methodology, I.N.Y., F.A.H. and I.S.; software, F.A.H.; validation, I.N.Y. and I.S.; formal analysis, I.N.Y.; investigation, I.N.Y. and A.S.P.; resources, F.A.H.; data curation, I.N.Y. and I.S.; writing—original draft preparation, I.N.Y., F.A.H. and A.S.P.; writing—review and editing, I.N.Y.; visualization, F.A.H.; supervision, A.S.P.; project administration, I.N.Y.; funding acquisition, I.N.Y. All authors have read and agreed to the published version of the manuscript.

Funding: This study was funded by the Higher Education Basic Research Program (Program Penelitian Dasar Unggulan Perguruan Tinggi/PDUPT), Ministry of Education and Culture with contract number 1207/UN6.3.1/PT.00/2021 and the research title "Cheating Detection in Online Exams Using Deep Learning Method".

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: The authors warmly appreciate the Rector, Directorate of Research and Community Service (DRPM), and the Research Center for Artificial Intelligence and Big Data at Universitas Padjadjaran. We also thank the World Class Professor (WCP) 2021 program, Ministry of Education and Culture, Indonesia.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Kumala, R.D.M. Legal Analysis of Government Policy on Large Scale Social Restrictions in Handling COVID-19. *Indones. J. Int. Clin. Leg. Educ.* **2020**, *2*, 181–200. [CrossRef]
2. Aboagye, E. Transitioning from face-to-face to online instruction in the COVID-19 era: Challenges of tutors at colleges of education in Ghana. *Soc. Educ. Res.* **2021**, *2*, 9–19. [CrossRef]
3. Putarek, V.; Pavlin-Bernardić, N. The role of self-efficacy for self-regulated learning, achievement goals, and engagement in academic cheating. *Eur. J. Psychol. Educ.* **2020**, *35*, 647–671. [CrossRef]
4. Ampuni, S.; Kautsari, N.; Maharani, M.; Kuswardani, S.; Buwono, S.B.S. Academic dishonesty in Indonesian college students: An investigation from a moral psychology perspective. *J. Acad. Ethics* **2020**, *18*, 395–417. [CrossRef]
5. Balderas, A.; Caballero-Hernández, J.A. Analysis of learning records to detect student cheating on online exams: Case study during COVID-19 pandemic. In Proceedings of the Eighth International Conference on Technological Ecosystems for Enhancing Multiculturality, Salamanca, Spain, 21–23 October 2020; pp. 752–757.
6. Abdelrahim, Y. How COVID-19 quarantine influenced online exam cheating: A case of Bangladesh University Students. *J. Southwest Jiaotong Univ.* **2021**, *56*, 1–10. [CrossRef]
7. Bilen, E.; Matros, A. Online cheating amid COVID-19. *J. Econ. Behav. Organ.* **2021**, *182*, 196–211. [CrossRef]
8. Rehman, A.; Saba, T.; Khan, M.Z.; Damaševičius, R.; Bahaj, S.A. Internet-of-Things-Based Suspicious Activity Recognition Using Multimodalities of Computer Vision for Smart City Security. *Secur. Commun. Netw.* **2022**, *2022*, 8383461. [CrossRef]
9. Dang, L.M.; Min, K.; Wang, H.; Piran, M.J.; Lee, C.H.; Moon, H. Sensor-based and vision-based human activity recognition: A comprehensive survey. *Pattern Recognit.* **2020**, *108*, 107561. [CrossRef]
10. Gu, F.; Chung, M.H.; Chignell, M.; Valaee, S.; Zhou, B.; Liu, X. A survey on deep learning for human activity recognition. *ACM Comput. Surv.* **2021**, *54*, 1–34. [CrossRef]

11. Bouchabou, D.; Nguyen, S.M.; Lohr, C.; LeDuc, B.; Kanellos, I. A survey of human activity recognition in smart homes based on IoT sensors algorithms: Taxonomies, challenges, and opportunities with deep learning. *Sensors* **2021**, *21*, 6037. [CrossRef]
12. Wang, P.; Fan, E.; Wang, P. Comparative analysis of image classification algorithms based on traditional machine learning and deep learning. *Pattern Recognit. Lett.* **2021**, *141*, 61–67. [CrossRef]
13. Khamparia, A.; Singh, K.M. A systematic review on deep learning architectures and applications. *Expert Syst.* **2019**, *36*, e12400. [CrossRef]
14. Ayachi, R.; Said, Y.; Atri, M. A convolutional neural network to perform object detection and identification in visual large-scale data. *Big Data* **2021**, *9*, 41–52. [CrossRef] [PubMed]
15. Nguyen, H. Improving faster R-CNN framework for fast vehicle detection. *Math. Probl. Eng.* **2019**, *2019*, 3808064. [CrossRef]
16. De-La-Hoz-Franco, E.; Ariza-Colpas, P.; Quero, J.M.; Espinilla, M. Sensor-based datasets for human activity recognition—A systematic review of literature. *IEEE Access* **2018**, *6*, 59192–59210. [CrossRef]
17. Novac, P.E.; Pegatoquet, A.; Miramond, B.; Caquineau, C. UCA-EHAR: A Dataset for Human Activity Recognition with Embedded AI on Smart Glasses. *Appl. Sci.* **2022**, *12*, 3849. [CrossRef]
18. Yulita, I.N.; Fanany, M.I.; Arymurthy, A.M. Combining deep belief networks and bidirectional long short-term memory: Case study: Sleep stage classification. In Proceedings of the 2017 4th International Conference on Electrical Engineering, Computer Science and Informatics, Yogyakarta, Indonesia, 19–21 September 2017; pp. 1–6.
19. Wang, J.; Chen, Y.; Hao, S.; Peng, X.; Hu, L. Deep learning for sensor-based activity recognition: A survey. *Pattern Recognit. Lett.* **2019**, *119*, 3–11. [CrossRef]
20. Wan, S.; Qi, L.; Xu, X.; Tong, C.; Gu, Z. Deep learning models for real-time human activity recognition with smartphones. *Mob. Netw. Appl.* **2020**, *25*, 743–755. [CrossRef]
21. Xu, C.; Chai, D.; He, J.; Zhang, X.; Duan, S. InnoHAR: A deep neural network for complex human activity recognition. *IEEE Access* **2019**, *7*, 9893–9902. [CrossRef]
22. Liu, H.; Lang, B. Machine learning and deep learning methods for intrusion detection systems: A survey. *Appl. Sci.* **2019**, *9*, 4396. [CrossRef]
23. Uddin, S.; Khan, A.; Hossain, M.E.; Moni, M.A. Comparing different supervised machine learning algorithms for disease prediction. *BMC Med. Inform. Decis. Mak.* **2019**, *19*, 281. [CrossRef]
24. He, H.; Jin, S.; Wen, C.K.; Gao, F.; Li, G.Y.; Xu, Z. Model-driven deep learning for physical layer communications. *IEEE Wirel. Commun.* **2019**, *26*, 77–83. [CrossRef]
25. Spoorthi, G.E.; Gorthi, S.; Gorthi, R.K.S.S. PhaseNet: A deep convolutional neural network for two-dimensional phase unwrapping. *IEEE Signal Process. Lett.* **2018**, *26*, 54–58. [CrossRef]
26. Yu, H.; Yang, L.T.; Zhang, Q.; Armstrong, D.; Deen, M.J. Convolutional neural networks for medical image analysis: State-of-the-art, comparisons, improvement and perspectives. *Neurocomputing* **2021**, *444*, 92–110. [CrossRef]
27. Khan, M.A.; Sharif, M.; Akram, T.; Raza, M.; Saba, T.; Rehman, A. Hand-crafted and deep convolutional neural network features fusion and selection strategy: An application to intelligent human action recognition. *Appl. Soft Comput.* **2020**, *87*, 105986. [CrossRef]
28. Arivazhagan, S.; Ligi, S.V. Mango leaf diseases identification using convolutional neural network. *Int. J. Pure Appl. Math.* **2018**, *120*, 11067–11079.
29. Patel, S. A comprehensive analysis of Convolutional Neural Network models. *Int. J. Adv. Sci. Technol.* **2020**, *29*, 771–777.
30. Saadati, M.; Nelson, J.; Ayaz, H. Mental workload classification from spatial representation of fnirs recordings using convolutional neural networks. In Proceedings of the 2019 IEEE 29th International Workshop on Machine Learning for Signal Processing (MLSP), Pittsburgh, PA, USA, 13–16 October 2019; pp. 1–6.
31. Rogelio, J.; Dadios, E.; Bandala, A.; Vicerra, R.R.; Sybingco, E. Alignment control using visual servoing and mobilenet single-shot multi-box detection (SSD): A review. *Int. J. Adv. Intell. Inform.* **2022**, *8*, 97–114. [CrossRef]
32. Gaba, S.; Budhiraja, I.; Kumar, V.; Garg, S.; Kaddoum, G.; Hassan, M.M. A federated calibration scheme for convolutional neural networks: Models, applications and challenges. *Comput. Commun.* **2022**, *192*, 144–162. [CrossRef]
33. Antelo, C.; Martinho, D.; Marreiros, G. A Review on Supervised Learning Methodologies for Detecting Eating Habits of Diabetic Patients. In Proceedings of the EPIA Conference on Artificial Intelligence, Lisbon, Portugal, 31 August–2 September 2022; pp. 374–386.
34. Li, Z.; Liu, F.; Yang, W.; Peng, S.; Zhou, J. A survey of convolutional neural networks: Analysis, applications, and prospects. *IEEE Trans. Neural Netw. Learn. Syst.* **2021**, *33*, 6999–7019. [CrossRef]
35. Atoum, Y.; Chen, L.; Liu, A.X.; Hsu, S.D.H.; Liu, X. Automated online exam proctoring. *IEEE Trans. Multimed.* **2017**, *19*, 1609–1624. [CrossRef]
36. Maharana, K.; Mondal, S.; Nemade, B. A Review: Data Pre-Processing and Data Augmentation Techniques. *Glob. Transit. Proc.* **2022**, *3*, 91–99. [CrossRef]
37. Mumuni, A.; Mumuni, F. Data augmentation: A comprehensive survey of modern approaches. *Array* **2022**, *16*, 100258. [CrossRef]
38. Shorten, C.; Khoshgoftaar, T.M. A survey on image data augmentation for deep learning. *J. Big Data* **2019**, *6*, 60. [CrossRef]
39. Nalepa, J.; Marcinkiewicz, M.; Kawulok, M. Data augmentation for brain-tumor segmentation: A review. *Front. Comput. Neurosci.* **2019**, *13*, 83. [CrossRef]

40. Moen, E.; Bannon, D.; Kudo, T.; Graf, W.; Covert, M.; Van Valen, D. Deep learning for cellular image analysis. *Nat. Methods* **2019**, *16*, 1233–1246. [CrossRef]
41. Zaidi, S.S.A.; Ansari, M.S.; Aslam, A.; Kanwal, N.; Asghar, M.; Lee, B. A survey of modern deep learning based object detection models. *Digit. Signal Process.* **2022**, *126*, 103514. [CrossRef]
42. Salih, A.A.; Abdulazeez, A.M. Evaluation of classification algorithms for intrusion detection system: A review. *J. Soft Comput. Data Min.* **2021**, *2*, 31–40. [CrossRef]
43. Saba, T.; Rehman, A.; Jamail, N.S.M.; Marie-Sainte, S.L.; Raza, M.; Sharif, M. Categorizing the students' activities for automated exam proctoring using proposed deep L2-GraftNet CNN network and ASO based feature selection approach. *IEEE Access* **2021**, *9*, 47639–47656. [CrossRef]
44. Malhotra, M.; Chhabra, I. Student Invigilation Detection Using Deep Learning and Machine After Covid-19: A Review on Taxonomy and Future Challenges. In *Future of Organizations and Work after the 4th Industrial Revolution: The Role of Artificial Intelligence, Big Data, Automation, and Robotics*; Springer: Cham, Switzerland, 2022; Volume 1037, pp. 311–326.
45. Mansour, R.F.; Escorcia-Gutierrez, J.; Gamarra, M.; Villanueva, J.A.; Leal, N. Intelligent video anomaly detection and classification using faster RCNN with deep reinforcement learning model. *Image Vis. Comput.* **2021**, *112*, 104229. [CrossRef]
46. Maschler, B.; Weyrich, M. Deep transfer learning for industrial automation: A review and discussion of new techniques for data-driven machine learning. *IEEE Ind. Electron. Mag.* **2021**, *15*, 65–75. [CrossRef]
47. Ayana, G.; Dese, K.; Choe, S.W. Transfer learning in breast cancer diagnoses via ultrasound imaging. *Cancers* **2021**, *13*, 738. [CrossRef] [PubMed]
48. Kumar, Y.; Gupta, S. Deep transfer learning approaches to predict glaucoma, cataract, choroidal neovascularization, diabetic macular edema, drusen and healthy eyes: An experimental review. *Arch. Comput. Methods Eng.* **2022**, *30*, 1–21. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.