11-1-2022

Towards effective and efficient online exam systems using deep learning-based cheating detection approach

Sanaa Kaddoura
Zayed University, sanaa.kaddoura@zu.ac.ae

Abdu Gumaei
Prince Sattam Bin Abdulaziz University

Intelligent Systems with Applications 16 (2022) 200153

Contents lists available at ScienceDirect

Intelligent Systems with Applications

journal homepage: www.journals.elsevier.com/intelligent-systems-with-applications

Towards effective and efficient online exam systems using deep learning-based cheating detection approach ☆

Sanaa Kaddoura [a,*], Abdu Gumaei [b]

[a] Zayed University, Abu Dhabi, United Arab Emirates
[b] Prince Sattam bin Abdulaziz University, Al-Kharj, Saudi Arabia

ARTICLE INFO

Keywords:
Deep convolutional neural networks
Cheating detection
Video frames
Speech
Soft voting decision fusion
Gaussian-based discrete Fourier transform

ABSTRACT

With the high growth of digitization and globalization, online exam systems continue to gain popularity and stretch, especially in the case of spreading infections like a pandemic. Cheating detection in online exam systems is a significant and necessary task to maintain the integrity of the exam and give unbiased, fair results. Currently, online exam systems use vision-based traditional machine learning (ML) methods and provide examiners with tools to detect cheating throughout the exam. However, conventional ML methods depend on handcrafted features and cannot learn the hierarchical representations of objects from data itself, affecting the efficiency and effectiveness of such systems. The proposed research aims to develop an effective and efficient approach for online exam systems that uses deep learning models for real-time cheating detection from recorded video frames and speech. The developed approach includes three essential modules, which constantly estimate the critical behavior of the candidate student. These modules are the front camera-based cheating detection module, the back camera-based cheating detection module, and the speech-based detection module. It can classify and detect whether the candidate is cheating during the exam by automatically extracting useful features from visual images and speech through deep convolutional neural networks (CNNs) and the Gaussian-based discrete Fourier transform (DFT) statistical method. We evaluate our system using a public dataset containing recorded audio and video data samples collected from different subjects carrying out several types of cheating in online exams. These collected data samples are used to obtain the experimental results and demonstrate the proposed work's efficiency and effectiveness.

1. Introduction

Online learning is on the rise and has rapid innovation worldwide due to significant demand from major world events. Massive open online courses (MOOCs) allow students who cannot contact the campus because of schedule or location constraints to enroll in online courses and access a wide range of educational resources. Students can take courses online using platforms anywhere in the wide world, so there is no need to come to campus in a typical classroom. Educators have a wide variety of multimedia content to deliver knowledge to students through online courses. In Liu et al. (2020), the authors stated that from 2008 to 2018, the number of students who took at least one online course increased by 151%. There are several critical components of any educational program that each educational institution must consider and know how to deal with, such as exams and assessment tests. The percentage of cheating students when taking online courses has become higher. Any educational institution needs to detect and prevent cheating to maintain its value to society. King and Case (2014) have concluded that the students' cheating percentage was rising in 2013, and more than 74% of students confirmed that it is easier to cheat in online courses than in typical courses. It cannot guarantee the prevention of cheating. The authors also clarified that about 29% of students were able to cheat through online courses because there is no possible way to allocate human proctors. In contrast, a human proctor can monitor the students throughout the exam that has been taken in a traditional and protected classroom environment. One of the main disadvantages of taking online exams is that it is difficult for providers of MOOCs to ensure that the students have well-learned the course material and covered the main knowledge areas of the course. There are some testing procedures for the institution to follow when they have online exams,

☆ This work is funded by Zayed University Research Incentive Fund (RIF) with grant number R20128.
* Corresponding author.
E-mail addresses: Sanaa.kaddoura@zu.ac.ae (S. Kaddoura), a.gumaei@psau.edu.sa (A. Gumaei).

https://doi.org/10.1016/j.iswa.2022.200153
Received 17 June 2022; Received in revised form 6 October 2022; Accepted 14 November 2022
Available online 17 November 2022

which start when the learners come to the certified testing center or campus to take the exam under the supervision of a human proctor. Some institutions do not mind if the students take exams anywhere as long as they are connected by the Internet. Such organizations depend on new tools such as ProctorU or Kryterion that allow students to access the exams anywhere (Kaddoura et al., 2022). They still need a remote proctor to monitor the students through a webcam during the test. In this case, the proctor needs to be trained to monitor the students remotely by carefully watching and listening for unusual behaviors of the test taker. The unusual behaviors that can stop the test or alert the test taker could be unusual movements of the eyes or disappear from the field view of the webcam. This paper aims to enhance the monitoring of online exams by introducing a multimedia analytics system to perform automatic and continuous online exam proctoring (OEP). We will provide real-time proctoring to detect the cheating behaviors of the test taker to enhance the integrity of online exams. To achieve our goals, we need to observe the audiovisual behaviors of the test takers to detect if there is any cheating behavior. In Reale et al. (2011) and Xiao et al. (2015), authors have been studying how to extract features from visual data and audio to analyze human behaviors. The approach in Atoum et al. (2017) studied how to monitor the test taker in the room using two cameras and a microphone. The proctor monitors two cameras; a webcam that focuses on the monitor's face or is set above the test taker; and a WearCam that could be attached to the test taker's eyeglasses to capture the field of view. The authors said that it is essential to monitor the sound surrounding the test taker, so the webcam comes with a microphone to record any sound in the exam room. They also said that the system should be able to detect actions such as cheating from the book, using the phone, searching the Internet, and detecting if another person is taking the exam on their behalf. Their system depends on a hybrid approach that combines two stages of the algorithm: extracting the features from the audiovisual streams and making decisions by taking the high-level features from the first stage output. The first stage consisted of six basic components to extract features indicative of cheating, including user verification, speech detection, text detection, active window detection, phone detection, and gaze estimation. The output of this stage was a binary number or probabilistic estimation of observing unusual behaviors. The output of the first stage was inputted to the next stage to end up with a joint decision across all components by extracting the high-level temporal features. In order to detect real-time cheating behavior, the new features were used to train and test a classifier to provide continuous real-time detection. A database of audio and vision was collected from 24 different subjects containing various types of math exam cheating behaviors to evaluate their system. The exam questions are multiple-choice and fill-in-the-blank. Their results showed the accuracy and efficiency of the system. However, their approach is based on handcrafted features and cannot learn the hierarchical representations of objects from the data, affecting the efficiency and effectiveness of online exam systems. In Kadam et al. (2021), the authors propose a lightweight technique for dealing with the image. Although this is a lightweight technique, we could not use a similar approach because the students' images should not be modified and should be considered as is to avoid false detection of cheating when the students did not commit cheating.

Many different approaches can be applied as well as future work. In Walambe, Marathe, et al. (2021), they used ensemble learning for lightweight object detection. This approach can be investigated to determine the opportunity to apply it in this work. In Walambe, Nayak, et al. (2021), the authors employ multimodal machine learning for stress detection. In Chaudhari et al. (2020), they used a promising approach for data augmentation using GANs. The same approach can be applied to modify the dataset and recheck the accuracy. In order to justify the outcomes of experiments, explainable AI can be used as in Joshi et al. (2021).

This research aims to develop an effective and efficient approach for online exam systems that uses deep learning models for real-time cheating detection from recorded video frames and speech. The developed approach includes three essential modules, which constantly estimate the critical behavior of the candidate student. These modules are the front camera-based cheating detection module, the back camera-based cheating detection module, and the speech-based detection module. It can classify and detect whether the candidate is cheating or not during the exam by automatically extracting useful features from visual images and speech through deep convolutional neural networks (CNNs) and the Gaussian-based discrete Fourier transform (DFT) statistical method. Below are the contributions of the proposed approach:

- The proposed approach is lightweight and takes into consideration front-camera, back-camera, and speech detection.
- A soft voting-based decision-level fusion rule is proposed to give different weights to the output scores from the modules based on their importance. The output class with the greatest sum of weighted probabilities gives the final target a cheating or non-cheating label.
- The proposed approach achieved high accuracy of cheating detection.

The rest of the paper is organized in the following sections. Section 2 explains the related works in the literature review. Section 3 presents the proposed approach. Section 4 explains the research methods. Section 5 describes the experiments and the evaluation results of the developed models. Section 6 is the conclusion and future work.

2. Related work

Over recent years, the demand for online learning has increased significantly. Researchers have tried to propose different methods to proctor online exams using various techniques. An online proctoring service will allow students to take their exams in their own space at home. This means that a real person will supervise them in real-time via webcam, microphone, and speakers. In Li et al. (2015), the authors proposed a framework for online proctoring depending on collaborative and automatic cheating behavior detection approaches using four different components. They have used a gaze tracker, two webcams, and an EEG sensor to build their hardware to monitor one cheating type, which is a reading answer from the papers. The first camera will monitor the tester's face and be placed above him, while the other one will monitor the subject's profile and be located on the subject's right-hand side. The work has to be developed more because this framework cannot detect various cheating behaviors and only focuses on one type of cheating. In Wahid et al. (2015), the researchers developed a web-based exam system to prevent cheating while taking online exams.

There is much research that focuses on monitoring online exams using human monitoring. Still, the main disadvantage is that it is very costly and needs many employees to monitor the exams. In Rosen and Carr (2013), the authors proposed a semi-automated computer machine proctoring using an intelligent desktop robot. The robot consists of 360-degree motion and camera sensors to record the video data. The data will then be transmitted to a monitoring center to detect any suspicious motion in the video. According to their results, the method has one main problem: it still cannot detect many cheating behaviors, such as when someone stands outside the camera's view. However, it can still see the test questions and provide the answers to the test-taker using pieces of paper or silent signals.

A secure browser is another way that allows students to take online exams in a secure environment where they are monitored through their mics and webcams to monitor their behavior to prevent students from using other computer resources to cheat. A fully digital Artificial Intelligence-Based Proctoring System (AIPS) is used without human monitoring while taking online exams. The system will record and analyze the students' behaviors during exams, and when the students try to cheat, the active system will flag such behavior and take action ac-

cordingly. Then the system will either generate a report or suspend the exam to be reviewed by the institution (OReilly & Creagh, 2016).

Beyond the educational field, authors started studying audiovisual behaviors in the multimedia community to monitor more types of behaviors. Authors in Xiao et al. (2015) proposed a model to monitor head motion in human interaction using audiovisual recordings. Moreover, in Nguyen et al. (2014), the authors extracted some cues from the audiovisual data to predict heritability in real job interviews. In contrast, authors in Lefter et al. (2013) detected various threats and aggression using audiovisual data, such as unwanted behaviors in public areas.

In Hussein et al. (2020), the authors classified online proctoring systems into three types: live proctoring systems, recorded proctoring systems, and automated proctoring systems. Live proctoring systems are primarily used in theoretical exams. It involves a human proctor to track eye movements, identify students' faces, and flag if students are found malpractice and cheating. The recorded proctoring system involves recording videos during online exams, and then the post-proctoring is used for face movements and tracking eyes, objects, face detection, and log analysis. Automated proctoring systems are the most complex and complicated ones to design. The system analyzed students' behaviors through various algorithms and technologies without any need for human proctoring.

Several existing online proctoring systems worldwide can be used to monitor online exam takers. ProctorU is one famous live OPS that uses a microphone and webcam to guide students and monitor them during online exams. The system mainly depends on the webcam, so students are requested to have an uninterrupted audiovisual connection throughout the exam session (Milone et al., 2017). The Kryterion system is similar to the ProctorU system, and both of them are not highly secure, and the institutions can't entirely depend on them to monitor their students (Prathish et al., 2016). A hybrid solution was recommended by one of the companies that rely on live professional trained proctors who monitor the exams and have the facility to interrupt the exam if they suspect something (Slusky, 2020).

XProctor is another popular OPS that relies on facial recognition, video streaming, and audio and photo graphics to track and monitor students. This system can be easily integrated with various LMS and can be installed on the student's computer (Slusky, 2020). Another famous OPS is TeSLa, which the European Commission funded as a tool that can be launched to combat cheating in online exams. The system was evaluated and tested by more than 18 different European universities, and it is one of the free-to-use authentication tools. It depends on the biometrics of the test takers, such as voice recognition, keystroke analysis, facial recognition, and fingerprint analysis, to detect if students in online exams are not cheating (Draaijer et al., 2017).

ProctorExam is one of the leading online proctoring services developed in Europe and requires less data collection than other systems. The institution has a choice to choose the type of supervision: either monitor in real-time or review after the session for greater flexibility. The system provides dual-view proctoring, including screen-sharing, a webcam, and a smartphone camera to view 360° of the testers' workspace to monitor everything. The Safe Exam Browser is computer software that turns any computer, temporarily, into a secure workstation. It consists of a browser and a kiosk application. The application will lock down the examination computer from browsing and exploring other applications and tabs. At the same time, the browser communicates with the quiz module of an LMS running on a server. It is a very secure software that monitors students who take their exams via unmanaged computers, like students' own laptops and tablets, and disables them from taking shortcuts and copying and pasting while taking their online exams (Slusky, 2020).

After reviewing some examples of various online proctoring systems, they depend on the number of necessary parameters selected based on hardware and ease of implementation accessible to the students. It could be a mic, camera, human proctor, gaze tracking, screen share/recording, biometrics, and application lock. The webcam is used to monitor students while concurrently recognizing any cheating attempts. It can be used to check other people in the background who are trying to help or support cheating. The mic can be utilized to record the audio speech to detect the background noises that tell whether the student is being supported via a call. Some software also depends on a human proctor, besides other parameters, because current systems do not have a 100 percent accuracy rate to detect online cheating and prevent a student from being wrongly accused. Screen sharing can be used in some applications to allow the proctor to view the students' tabs and monitor their screen equipment to ensure they did not open any other tabs or notes to search for the answer. In contrast, recordings can be used in some applications as a reference if there is a disagreement about the suspicious activity flag raised by the system (Nigam et al., 2021).

Application lock is one of the parameters that can be used to lock the application or the browser while the student is taking online exams to ensure that students cannot get access to other software in the exam background. This parameter is used via a secure browser method. It flags the students when creating a tab switch. Biometrics parameters use biometric verification to ensure that the student is not cheating by impersonating someone else. The last parameter is gaze tracking, which depends on a gaze tracker to prevent students from cheating using external resources. It monitors their behaviors when looking away from the screen (Nigam et al., 2021).

Based on recent research, most educational institutions have begun to embrace online proctoring software to conduct remote examinations while maintaining the integrity of the exam. Shifting from the traditional paper-based method to a remote proctoring system has many advantages. The first advantage is that scheduling exams become easier and make education more accessible worldwide. Students can take the exam from any place at any time. Moreover, student-instructor communication becomes faster, and the exams' results can be generated in a shorter time (Lee & Fanguy, 2022).

By developing a multi-modal system, Malhotra et al. (2022) described a strategy for avoiding the physical presence of a proctor during the test. They used a webcam and active window capture to capture video. The test taker's face is recognized and analyzed to predict his emotions. The head pose is determined by identifying some feature points. A mobile phone, a book, or the attendance of another person are among the things that can be detected. This model combination yields an intelligent rule inference system capable of determining whether any examination malpractice occurred.

Noorbehbahani et al. (2022) introduced a review of 58 online cheating publications published between January 2010 and February 2021. They presented the trending topics about cheating on online exams. Their research can serve as a useful resource for educators and researchers working in online learning who want to get a more comprehensive understanding of cheating detection, mitigation, and prevention.

To ensure the integrity of the user, Gopane and Kotecha (2022) proposed a methodology that includes continuous user verification and validation. Eye gaze tracking and subtle expression detection, such as laughter detection to predict the applicant's viewing direction, eyes closing or blinking duration, and head movement and activity were detected for monitoring during the test. Any suspicious activity by the applicant was monitored and assessed. Artificial intelligence was used to classify the applicant's activities. The preliminary results showed that the proposed method was effective in this regard.

Sapre et al. (2022) proposed an intelligent online framework for reducing student malpractices in the online exam mode. They used some machine learning algorithm-based methods. During the exam, the main task was to continuously detect each student, which necessitated using a webcam for facial detection. To be sure, facial illumination and a proper pose setup were observed. The examiner can track the student from their end and receive alerts based on their illegal behavior because of the live detection.

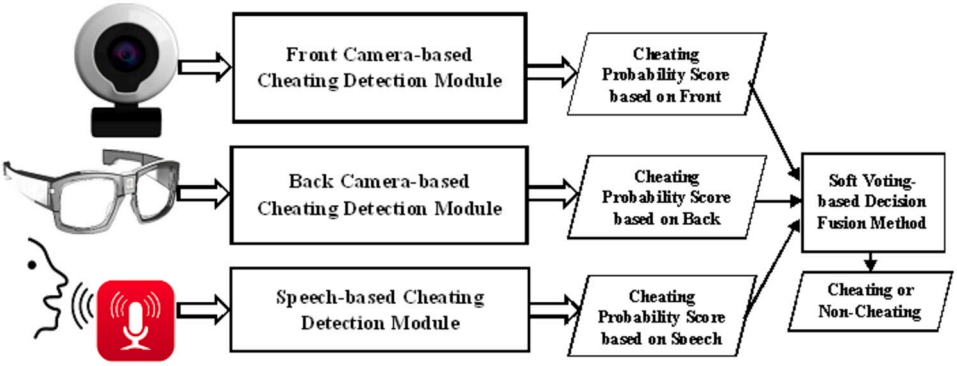

Fig. 1. The diagram of the proposed approach.

Jalali and Noorbehbahani (2017) proposed an automatic cheating detection method for analyzing the webcam images of students in online exams using image-based thresholding and clustering methods. The proposed method has been put to the test on actual students, and the evaluation results indicate that it can be used successfully in online exams. The average accuracy score across ten students is 78%. However, the accuracy score has still not improved, and there is no module to analyze students' sounds during the exam. Masud et al. (2022) proposed a method to detect cheating through detecting activity from exam videos by extracting four different types of event data and using a pre-trained classification model. The method transforms each video into a multivariate time series representing the time-varying event data. Several experiments on a real dataset of cheating videos showed that the method could predict outcomes with an accuracy of up to 97.7%. However, the multivariate time series requires a large data sample; otherwise, the method's results will be meaningless due to high standard errors. Moreover, multivariate features are complex and require high-level mathematical calculations.

Through the literature review and related work, it is clear that researchers have made several attempts to develop high-performance cheating detection systems. These attempts are either expensive or not supported by the results or have not reached the required efficiency and strength. Therefore, we focus on breaking the gap and limitations found in previous studies through this research.

3. Proposed approach

The proposed approach detects cheating in online exam systems based on effective and efficient methods with the decision fusion rule. It mainly consists of three detection modules that give cheating probabilities scores and fuse them using a soft voting technique, to be described in the following subsection. Fig. 1 demonstrates the diagram of the approach. The first and second modules take a sequence of video frames, depending on a selected time window size, captured from front and back cameras as inputs. A lightweight end-to-end deep convolutional neural network (CNN) method classifies them into cheating or non-cheating classes according to their probability scores. The third module takes chunks of recorded voice according to the same window size as the first and second modules. Then, it classifies them into: cheating class, for the speech occurrence with one probability score, or non-cheating class, for non-speech occurrence with one zero probability score using a Gaussian-based discrete Fourier transform (DFT) statistical method introduced in Sohn et al. (1999). For speech detection, the Gaussian statistical coefficient vectors of noise, speech, and noisy speech with kth elements are taken. After that, a soft voting-based decision fusion method (Gumaei et al., 2022) gives different weights to the output scores from the modules based on their importance and sums them up. Then, the output class, with the greatest sum of weighted probabilities, obtains the vote for the final target cheating or non-cheating label. The idea behind using a sequence of images and chunks of voice, based on the time window size, is to maintain the compulsory requirement of the space. Thus, only the sequence of images and chunks of voice for the cheating activity will be saved for reviewing the behavior of any candidate during the online exam if this is needed in some cases.

4. Research methods

In this section, the methods used in the proposed approach are explained. The deep CNN method used in the first and second modules for cheating detection from video frames captured by front and back cameras is described first. Two models are built: one model detects cheating from front camera images, and the second model detects cheating from back camera images. Then, the Gaussian-based DFT statistical method for cheating detection from the speech is described. Finally, the soft voting decision level fusion method will be explained. The following subsections describe these methods in detail.

4.1. A lightweight deep convolutional neural network (CNN) method

Fig. 2 shows the block diagram of the proposed deep CNN model architecture. It contains five major blocks that extract and classify cheating features from input images. The blocks numbers one to four consist of four 2D convolutional layers and four max-pooling layers. Two dropout layers are used in blocks four and five to prevent the overfitting problem during the model's training. Block number 5 has a series of fully connected (FC) layers: the flattening layer and two dense layers separated by the dropout layer.

The model's input images are resized to be 224×224×3, where 224 is the size of both height and width, and 3 is the number of color channels for the layers. A rectified linear unit (ReLU) is employed as an activation function in each layer of the model. The ReLU is a non-linear and simple function that makes a large network easy to train. The following equation (Gumaei et al., 2020) can be used to calculate it.

$$\sigma(u) = max(0, u) \tag{1}$$

The kernels used in the convolution model are sized to have a fixed size of 2×2, and the max-pooling is also sized to 2×2 to reduce the number of parameters and computation process. For the input images, the CNN layers can provide various levels of feature abstraction. Fig. 2 displays the total number of features created.

In the final block, the classification process is accomplished. The fourth step generates 2D feature maps, which are then flattened into a 1D feature vector. The 1D feature vector is fed into a dense layer with completely connected layers in the fifth block. The dense layer's output is then transferred through a dropout layer before being transmitted through another dense layer with a softmax activation function. It has several neurons equal to the number of class labels. The likelihood of each class label is calculated using the softmax activation function. It can be calculated using the following formula (Gumaei et al., 2020):

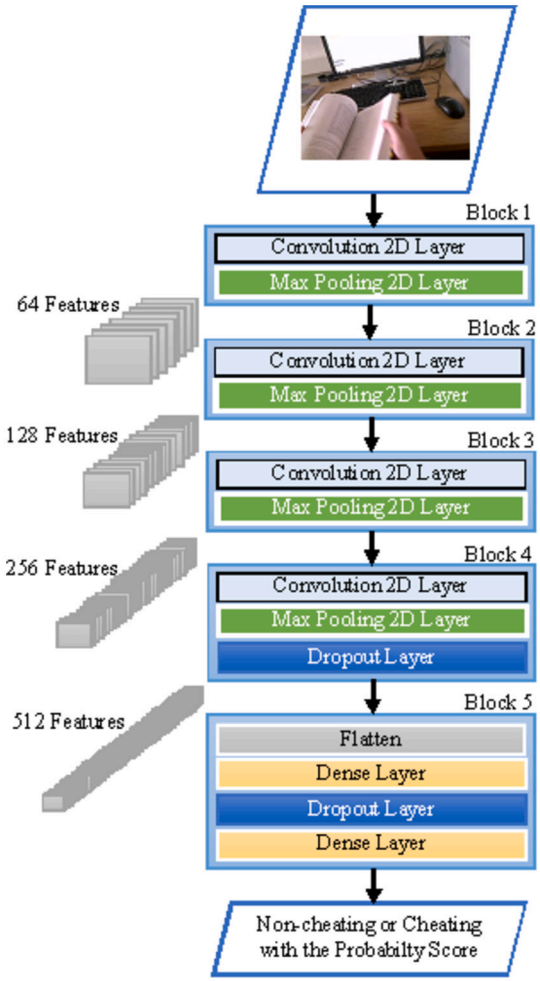

Fig. 2. Block diagram of custom DCNN model.

$$\text{prob}_k = \frac{e^{u_k}}{\sum_{i=1}^{n} e^{u_i}}, \text{ for } k = 1, \ldots, n \tag{2}$$

where *prob* is the output probability between zero and one for the input u, and the sum of the outputs is equal to one.

This model has few layers and a small number of learnable parameters. Therefore, it is lightweight and has a low detection computation cost.

4.2. A Gaussian-based discrete Fourier transform (DFT) statistical method

The method mentioned in Sohn et al. (1999) is used for speech-based cheating detection. Suppose the task is to detect the speech from recorded voices, and there is uncorrelated additive noise coming from the background environment; the two hypotheses to detect the speech that can be considered for each chunk of voice are:

H_0: There is no speech, $Y = N$

H_1: There is a speech, $Y = S + N$

Here N, S, and Y are the coefficient vectors of the discrete Fourier transform (DFT) for noise, speech, and noisy speech with their jth elements N_j, S_j and Y_j respectively. The Gaussian statistical model is adopted in which each process generates the DFT coefficients that are independent asymptotically Gaussian random variables (Ephraim & Malah, 1984). At this point, the probability density functions (PDFs) of H_0 and H_1 are computed by:

$$\text{Prob}\left(Y \mid H_0\right) = \prod_{j=0}^{N-1} \frac{1}{\pi\sigma_N(j)} \exp\left\{\frac{\left|Y_j\right|^2}{\sigma_N(j)}\right\} \tag{3}$$

$$\text{Prob}\left(Y \mid H_1\right) = \prod_{j=0}^{N-1} \frac{1}{\pi\left[\sigma_S(j) + \sigma_N(j)\right]} \exp\left\{\frac{\left|Y_j\right|^2}{\sigma_S(j) + \sigma_N(j)}\right\} \tag{4}$$

where $\sigma_S(j)$ and $\sigma_N(j)$ are the variances of S_j and N_j, respectively. The ratio of the likelihood for the jth frequency bands is:

$$\mathcal{L}_j \triangleq \frac{\text{Prob}\left(Y_j \mid H_1\right)}{\text{Prob}\left(Y_j \mid H_0\right)} = \frac{1}{1+\mathfrak{J}_j} \exp\left\{\frac{\gamma_j \mathfrak{J}_j}{1+\mathfrak{J}_j}\right\} \tag{5}$$

where $\gamma_j \triangleq \left|Y_j\right|^2 / \sigma_N(j)$ and $\mathfrak{J}_j \triangleq \sigma_S(j)/\sigma_N(j)$ are called a posteriori and a priori for the signal-to-noise ratio (SNR), respectively (Reale et al., 2011). The decision rule is derived from the geometric mean of likelihood ratios for each frequency band, which is calculated as follows:

$$\log \mathcal{L} = \frac{1}{N} \sum_{i=0}^{N-1} log\mathcal{L}_1 \underset{H_0}{\overset{H_1}{\gtrless}} \eta \tag{6}$$

Through the noise statistic estimation procedure, we consider that $\sigma_N(j)$'s are already known, and the unknown parameters, $\mathfrak{J}_j$'s need to be estimated.

The maximum likelihood (ML) estimator for $\mathfrak{J}_j$ can simply be obtained using the following equation:

$$\hat{\mathfrak{J}}_j^{(\text{ML})} = \gamma_j - 1 \tag{7}$$

By substituting (7) into (6) and after applying the likelihood ratio test (LRT) produces the Itakura–Saito distortion (ISD) based on the decision rule (Sohn & Sung, 1998), as follows:

$$\log \hat{\mathcal{L}}_j^{(\text{ML})} = \frac{1}{m} \sum_{i=0}^{m-1} \left[\gamma_j - \log \gamma_j - 1\right] \underset{H_0}{\overset{H_1}{\gtrless}} \eta \tag{8}$$

The left-hand side of (8) cannot be less than zero, which is a well-known characteristic of ISD and indicates that the likelihood ratio is skewed toward H_1. We use the DD a priori SNR estimate technique (Ephraim & Malah, 1984) to mitigate this bias:

$$\hat{\mathfrak{J}}_j(m)^{(DD)} = \alpha \frac{\hat{\mathcal{L}}_j^2(m-1)}{\sigma_N(j, m-1)} + (1-\alpha)\,\text{Prob}\left[\gamma_j(m) - 1\right] \tag{9}$$

where m is the frame-index and the $Prob(u) = u$, if $u \geq 0$, otherwise, $Prob(u) = 0$ and $\mathcal{L}_j(m-1)$'s are the previous frame signal amplitude estimates, in which the minimum mean square error (MMSE) estimator is used (Ephraim & Malah, 1984). The DD method in (9) delivers the priori SNR smoother estimates than the ML technique (Cappé, 1994), reducing the variability of predicted likelihood ratios throughout the noise-only periods.

4.3. Soft voting-based decision fusion method

The method used is the fusion method, which uses a soft voting strategy to merge practically diverse probability scores to get the final decision output. This strategy is recommended when the probability scores are well calibrated, and diverse (Karlos et al., 2020). The soft voting-based method classifies the class label based on the probabilities generated by several classification members. Different weights are applied to each classification member based on its importance using the equation given below:

$$z = \arg\max_i \sum_{j=1}^{n} w_j p_{i,j} \tag{10}$$

where w_j is the weight of each classification member and $p_{i,j}$ denotes the output probability score of the class label i and the classifier j.

Assume that the cheating classification task is formulated as a binary classification where the non-cheating class is labeled 0 and the cheating class is labeled 1. Furthermore, we assume that the probabilities of classification for an example are produced by three decision levels (D_1, D_2, D_3) are given as:

$$D_1(u) = [prob_{0,1}, prob_{1,1}] \tag{11}$$

$$D_2(u) = [prob_{0,2}, prob_{1,2}] \tag{12}$$

$$D_3(u) = [prob_{0,3}, prob_{1,3}] \tag{13}$$

In our case, D_1 is the decision probability of front camera cheating detection, D_2 is the decision probability of the back camera, and D_3 is the decision probability of speech detection. By using different weights, the sum-up of probabilities with weights for the final decision can be calculated as follows:

$$prob(l = 0, u) = w_1 p_{0,1} + w_2 p_{0,2} + w_3 p_{0,3} \tag{14}$$

$$prob(l = 1, u) = w_1 p_{1,1} + w_2 p_{1,2} + w_3 p_{1,3} \tag{15}$$

$$z = \arg\max_{l}[prob(l = 0, u), prob(l = 1, u)] \tag{16}$$

The value of z is the final decision output of the proposed soft voting decision fusion method.

5. Experiments and discussion

This section explains the experimental results and evaluates the proposed approach on two datasets collected from a public database using several evaluation metrics to validate its applicability. The experiments have been carried out on a laptop with a 2.2 GHz Intel Core i7-8750 processor, an 8 GB NVIDIA GEFORCE GTX display card, and 32 GB of RAM with a 64-bit Windows 10 operating system and used in the Python programming language through the KERAS library. The datasets and evaluation metrics with the results and discussion are given in the following subsections.

5.1. Datasets description

The datasets used in the experiments were collected from the public online exam proctoring (OEP) database (Atoum et al., 2017). The OEP database videos and audio were collected using a webcam (front camera), a WearCam (back camera), and an integrated microphone. It has 24 individuals, and all are Michigan State University undergraduates. Actors played 15 subjects who pretended to be doing the exam. They were instructed to engage in cheating activities throughout the session without being told what they should do or how they should do it. One concern with these individuals is that they exhibit potentially faked behaviors while acting. Therefore, nine students are requested to take a real exam and record their results to capture real-world exam conditions. These nine subjects from the 24 subjects are Subjects 10–16, Subject 18, and Subject 19. The proctor instigates cheating by conversing, handing them a book, stepping up to the student, etc., knowing that they are unlikely to cheat in the capturing room of data collected. When these two subjects are combined, the database is enriched with various cheating strategies and an engaging sense of real tests. The audio files that contained the audio information during the exam were named by the candidate's username. Therefore, we use the username as the name of the audio (.wav) file. For training and testing the approach, two datasets of images (Front-Cam dataset and Back-Cam dataset) with audio are created from this database. The first dataset is for detecting cheating from the front camera video frames, and the second dataset is for detecting cheating from the back camera video frames. Because processing and annotating the extracted frames of videos need high processing power and is time-consuming, the video with audio files of 12 subjects (Subjects 1–5, Subjects 18–24) is selected. Then, the video frames are converted into images. After visualizing the images, the keyframes are selected and annotated manually to create the two datasets. The number of images in the two datasets for non-cheating and cheating class labels is shown in Fig. 3. Some non-cheating and cheating samples taken from the datasets of images are shown in Fig. 4. We rename the audio files to have the ID of the subject. For example, the voice of subject 1 is "1.wav". The time duration of the sounds in the selected audio files is different, and it ranges between 13.46 and 25.07. Of the 12 audio files, only audio files of subjects 18, 19, and 20 contain a speech during the exam.

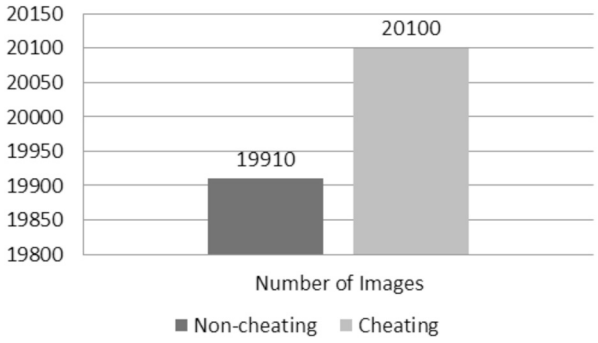

Fig. 3. The number of images in the two datasets.

5.2. Evaluation metrics

In this subsection, we give the performance evaluation metrics of the proposed approach, which are the accuracy, precision or Positive Predictive Value (PPV), recall or True Positive Rate (TPR), and False Positive Rate (FPR). These metrics can be calculated using the following equations.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{17}$$

$$Precision(PPV) = \frac{TP}{TP + FP} \tag{18}$$

$$Recall(TPR) = \frac{TP}{TP + FN} \tag{19}$$

$$FalsePositiveRate(FPR) = \frac{FP}{FP + TN} \tag{20}$$

where TP indicates the true positives, the number of correctly detected. FP represents the false positives, the number of incorrectly detected. TN represents the true negatives, the number of correctly detected. FN represents the number of false-negative occurrences that are incorrectly detected. The precision is the ratio of true positives to the sum of true positives and false positives. The ratio of true positives to the sum of true positives and false negatives is known as recall (also known as sensitivity). The FPR is the error in binary classification wherein a classification result incorrectly indicates the presence of cheating when the cheating is not present.

5.3. Experimental results and discussion

In the experiments, the hyper-parameters of the deep CNN model are initialized to their best values. Choosing the best hyper-parameter value in deep learning is a difficult task. As a result, we begin with a wide range of values and then narrow it down based on the validation results. The RMSprop optimizer was chosen to optimize the training process for the two models because it is a fast and widely used optimizer. The RMSprop optimizer's learning rate and rho parameters are left at their default values. The first fully connected dense hidden layer contains 500 neurons, while the second fully connected dense output layer contains one neuron to give a binary output with a probability score. A value of 0.5 is also set for the dropout ratio. The convolutional layers' kernel size is set to two, with a maximum pooling size of two, and the number of epochs is set to 30 for training the model.

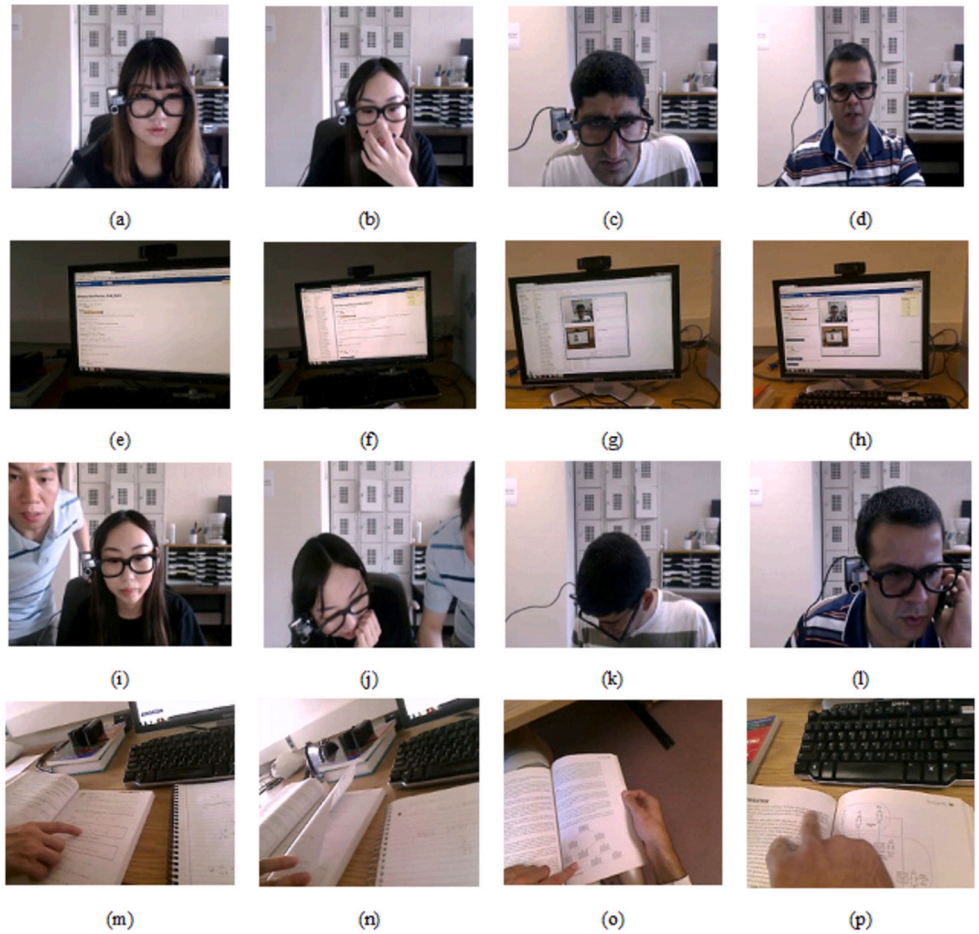

Fig. 4. Some samples taken from the OEP database videos [5]: (a)-(h) are for non-cheating class captured from front and back camera videos; (i)-(p) are for cheating class captured from front and back camera videos.

Two types of evaluation analysis have been performed to evaluate the flexibility of the proposed approach: evaluation with 30% of datasets for testing and evaluation with 40% of datasets for testing. In the first type, the datasets are divided into 50% for training, 20% for validation, and 30% for testing. For the second evaluation type, the models are trained on 40% of the datasets, validated on 20%, and tested on 40%. After that, the accuracy and other evaluation metrics are calculated. The number of instances in the training, validation, and test sets of the first evaluation type is 22405, 5602, and 12003 images. Also, the number of instances in the training, validation, and test sets of the second evaluation type is 19204, 4802, and 16004 images, respectively. By using these evaluation types and the evaluation of speech detection, the experimental results are divided into three groups in the following subsections:

5.3.1. Results on front and back camera-based cheating detection

The front camera-based cheating detection module results are presented in the following tables and figures. Fig. 5 shows the accuracy, loss of training, and validation sets resulting from the deep CNN models trained on 50% and validated on 20% of the datasets during training epochs. Figs. 5 (a) and (b) are the accuracy and loss for training and validation of the front camera model. Similarly, Figs. 5 (c) and (d) are for the back camera evaluation model.

Fig. 5 shows the stability of the training process of the developed model in which the values of accuracy are high, and the values of loss are small. In addition, there is no gap between the training and validation accuracy. This means that there is no overfitting in training the model.

Fig. 6 shows the confusion matrices of classified test sets obtained from the deep CNN model. Figs. 6 (a) and (b) are the confusion matrices for testing the deep CNN model of the front camera experiment on 30% test set size and 40% test set size. Similarly, Figs. 6 (c) and (d) show the confusion matrices for testing the model of the back camera experiment on 30% test set size and 40% test set size.

Confusion matrices provide a complete view of how the models are performing. They allow the computing of the other classification measures, which can guide model selection. According to these confusion matrices in Fig. 6, the accuracy rate reached by the developed lightweight deep CNN model is 99.83% for the front camera experiment on 30% test set size, and the accuracy rate achieved on 40% test set size is 99.81%. Furthermore, for the back camera experiment on 30% test set size and 40% test set size, the accuracy rates are 98.78% and 98.78%, respectively.

Tables 1 to 4 list the results of other evaluation metrics on the test sets. The proposed models achieve high recall, precision, and accuracy for the front and back-based modules. In addition, the models have low values of FPR (error in binary classification) for both non-cheating and cheating labels. Moreover, the developed models can effectively detect the subject-independent test images, which are not in the validation and training sets. These experimental results show the effectiveness and applicability of developed deep CNN models for building the proposed approach.

5.3.2. Results on speech-based cheating detection

In order to evaluate the utility of the speech-based cheating detection module, classification rates are computed for each subject audio

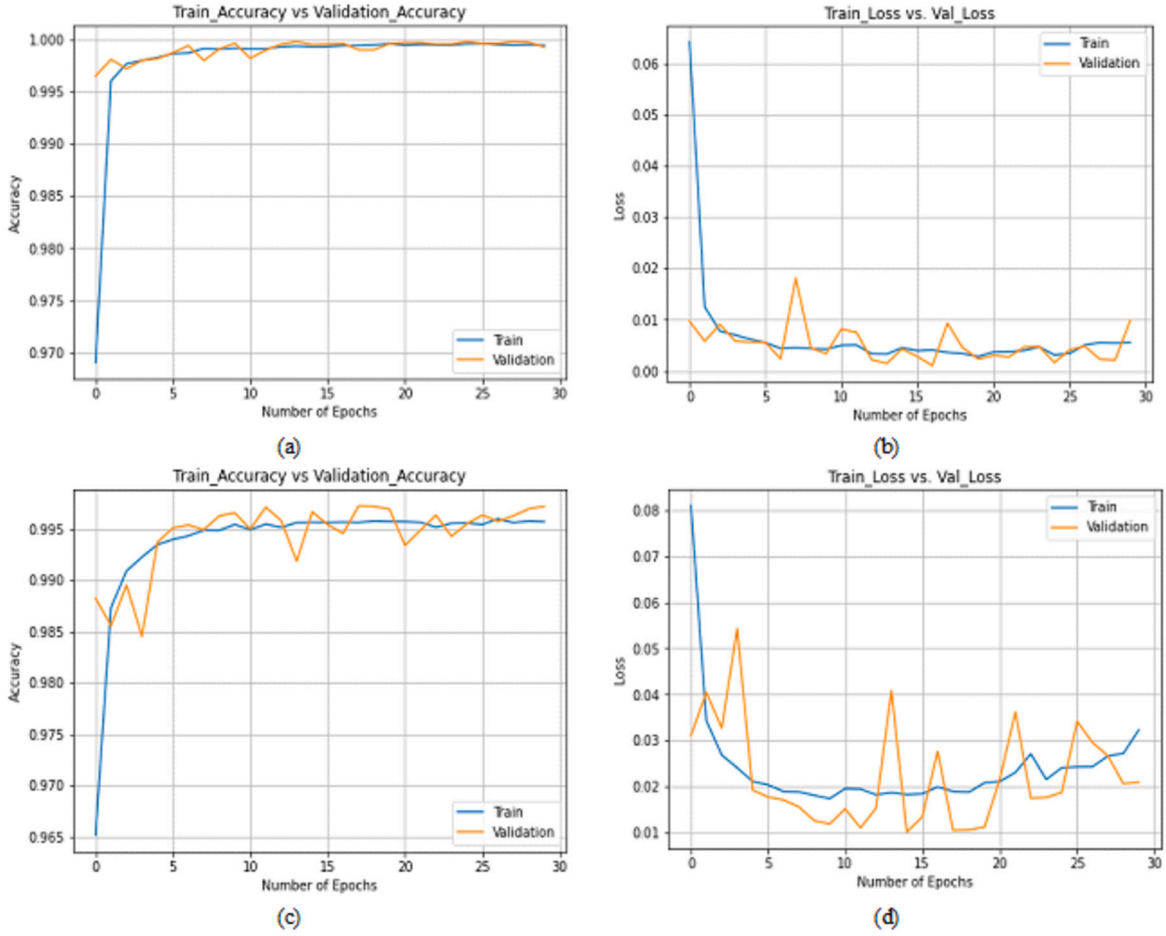

Fig. 5. Training progress of developed deep CNN model: (a) and (b) accuracy and loss for training and validation of front camera model; (c) and (d) accuracy and loss for training and validation of back camera model.

Table 1
Experimental evaluation results of deep CNN model for front camera experiment on 30% test set size.

Class Name	Recall	Precision	False Positive Rate
Non-Cheating	0.9973	0.9992	0.001
Cheating	0.9992	0.9974	0.003
Accuracy		99.83%	

Table 2
Experimental evaluation results of deep CNN model for front camera experiment on 40% test set size.

Class Name	Recall	Precision	False Positive Rate
Non-Cheating	0.9982	0.9979	0.002
Cheating	0.9979	0.9983	0.002
Accuracy		99.81%	

Table 3
Experimental evaluation results of deep CNN model for back camera experiment on 30% test set size.

Class Name	Recall	Precision	False Positive Rate
Non-Cheating	0.9955	0.9803	0.02
Cheating	0.9803	0.9955	0.005
Accuracy		98.78%	

file in the collected dataset (Subjects 1–5, Subjects 18–24) using the decision rule published by Sohn et al. (1999). The classification results of speech detection for the mentioned subjects are shown in Figs. 7 and 8.

Table 4
Experimental evaluation results of deep CNN model for back camera experiment on 40% test set size.

Class Name	Recall	Precision	False Positive Rate
Non-Cheating	0.9911	0.9844	0.015
Cheating	0.9845	0.9911	0.009
Accuracy		98.78%	

As shown in Fig. 7, there is no speech for subjects 1–5 rather than background noise, and the method detects that there is no speech and maintains the outputs as a vector of zeros. In Fig. 8, it is obvious that the speech module method can detect a speech during the exam in the recorded audio files of subjects 18, 19, and 20, and there is no speech for subjects 21–24. The method gives one output for the portions or chunks of voice that contain speech. There are three files of the 12 subjects on the audio files containing a speech, and nine files do not contain a speech. Effectively and correctly, the method detects all of them with 100% accuracy (i.e., "accuracy" = (3 + 9)/12).

5.3.3. Computational cost

In this section, the computational cost of the proposed approach is computed to confirm its applicability for real-time cheating detection in the online exam. The average detection time of the Deep CNN and speech cheating detection models during the experiments is presented in Table 5 to show how long they take to detect an unknown sample and a speech in the second time window. Consequently, it shows that the two models have a small computational cost that makes them efficient for online exam real-time cheating detection systems.

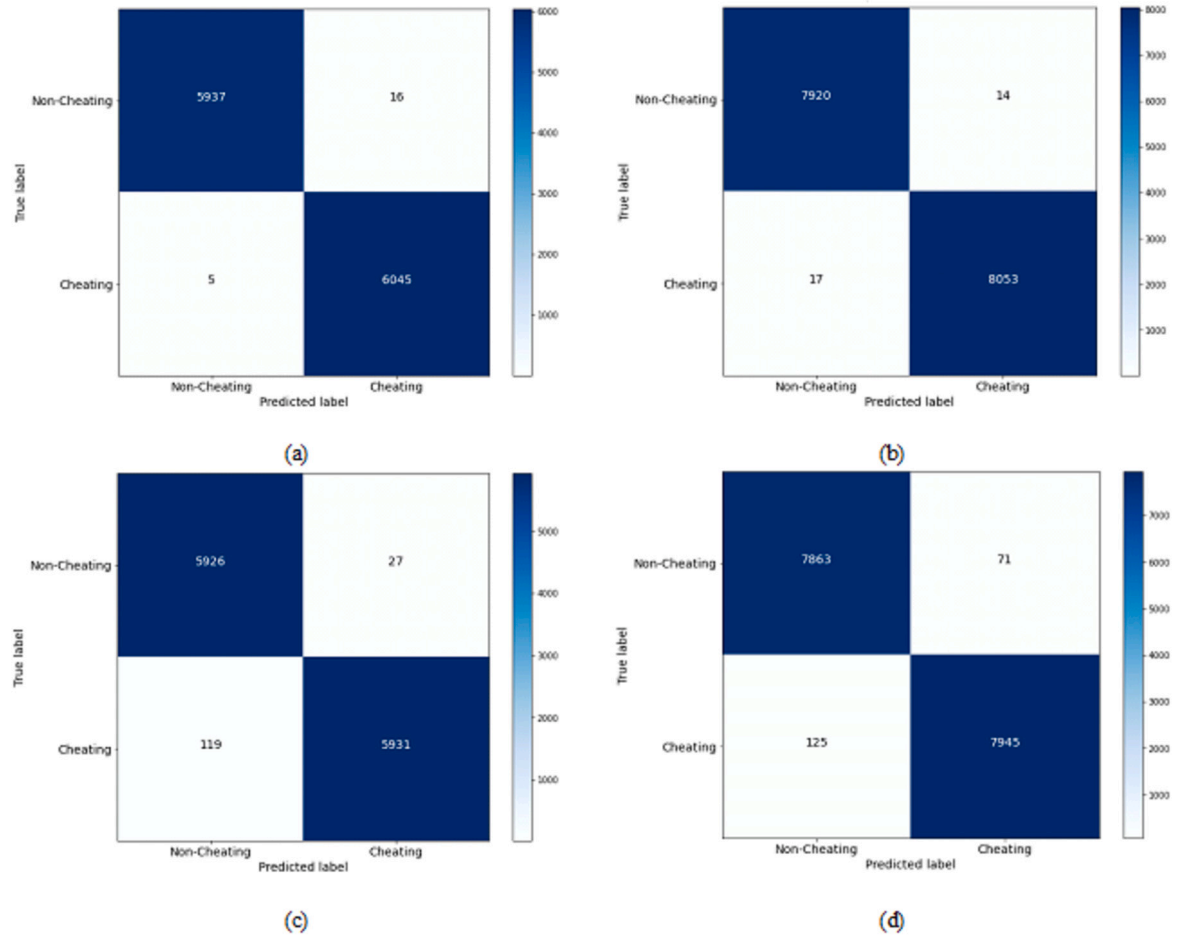

Fig. 6. Confusion matrices of the developed deep CNN model: (a) and (b) are the confusion matrices for testing the model of the front camera experiment on 30% test set size and 40% test set size; (c) and (d) are the confusion matrices for testing the model of the back camera experiment on 30% test set size and 40% test set size.

Table 5
Average time of the two models for detecting one sample image and a second of voice.

Method	Detection Time in Second
Deep CNN-based Cheating Detection per Sample	0.028
Speech-based Cheating Detection per Second of Voice	0.082

Table 6
The accuracy of the proposed approach against the current state-of-art methods.

Reference	Accuracy
Jalali and Noorbehbahani (2017)	78%
Masud et al. (2022)	97.7%
This Work	99.83%

5.3.4. Comparison of accuracy results

To validate the effectiveness of obtained results, we compared the proposed approach with the current state-of-art methods. Table 6 illustrates the accuracy of this work against the accuracies of related studies in Jalali and Noorbehbahani (2017), Masud et al. (2022).

Table 6 shows that the accuracy of the developed approach for detecting cheating and non-cheating classes is 99.83%, which is higher than the accuracy achieved by the proposed methods in Jalali and Noorbehbahani (2017), Masud et al. (2022). The comparison demonstrates the effectiveness of the proposed approach and confirms that it outperforms the current state-of-the-art methods.

In Jalali and Noorbehbahani (2017), the average accuracy score across ten students is 78%. However, this work lacks a module to analyze students' sounds during the exam. In Masud et al. (2022), the authors transformed each video into a multivariate time series representing the time-varying event data. Although the accuracy is 97.7%, the multivariate time series requires a large data sample; If a large sample is not used, the method's results will be meaningless due to high standard errors. Moreover, multivariate features are complex and require high-level mathematical calculations.

6. Conclusion and future work

In this research, a combination of computer vision methods and deep learning models were used to detect cheating in an online exam, with constructive results. However, an effective and lightweight detection approach for cheating detection is still needed, as it is a powerful and practical training module. This paper proposed a lightweight approach for real-time cheating detection based on deep CNN and Gaussian-based DFT statistical methods with decision fusion. The approach mainly consists of three detection modules that give cheating probabilities scores and fuse them using a soft voting technique. Extensive experiments on a publicly large-scale database were conducted to evaluate the proposed approach based on several evaluation metrics. Two types of evaluation analysis have been performed: evaluation with 30% of datasets for testing and evaluation with 40% of datasets for testing. In the first type, the datasets are divided into 50% for training, 20% for validation, and

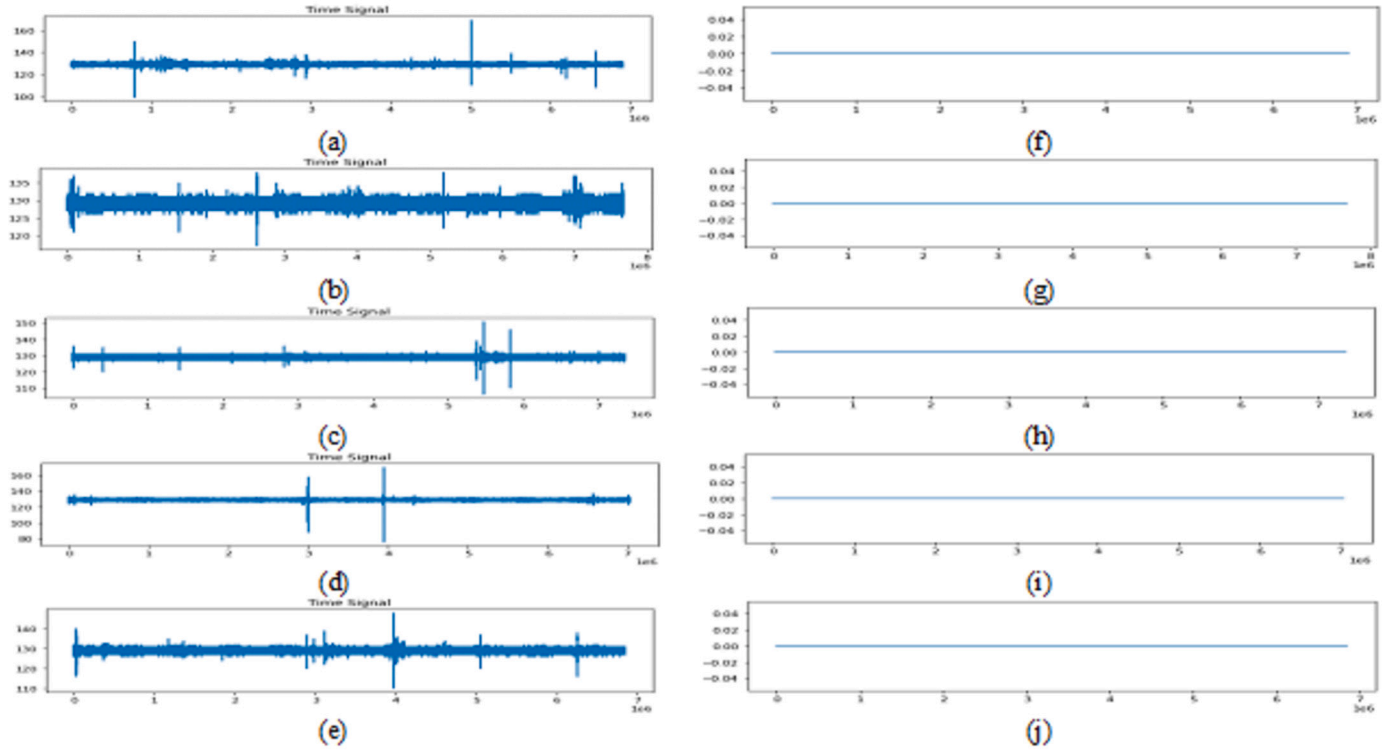

Fig. 7. Speech detection of subjects 1-5: (a)-(e) time signal of audio files; and (f)-(j) detection outputs through the whole of each audio file.

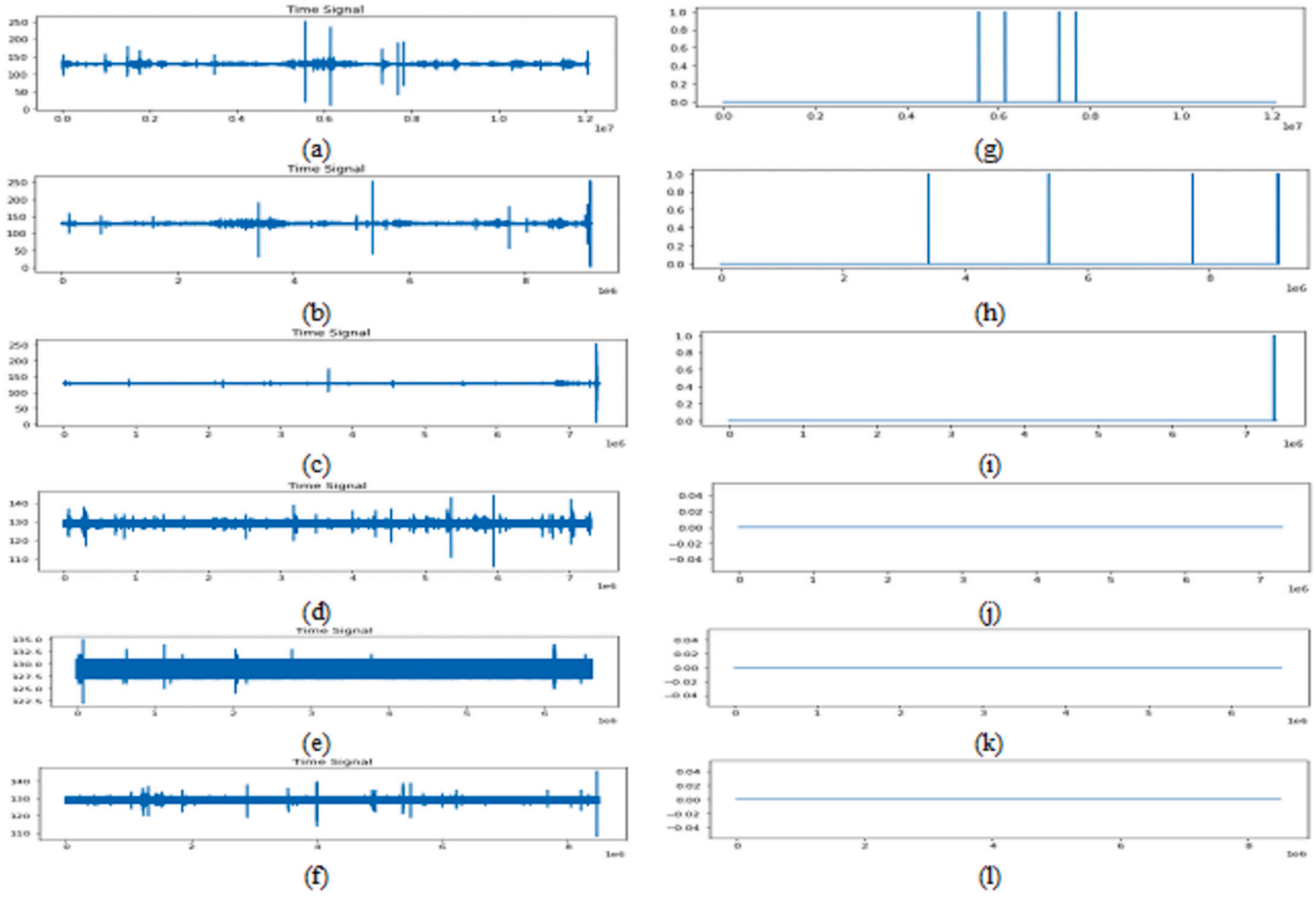

Fig. 8. Speech detection of subjects 18-24: (a)-(f) time signal of audio files; and (g)-(l) detection outputs through the whole of each audio file.

30% for testing. For the second evaluation type, the models are trained on 40% of the datasets, validated on 20%, and tested on 40%. The accuracy rates achieved by the developed front camera deep CNN model are 99.83% and 99.81% on 30% test set size and 40% test set size. Furthermore, the accuracy rates for the back camera deep CNN model are 98.78% and 98.78% on 30% test set size and 40% test set size, respectively. The computational costs of the proposed approach's methods are 0.028 seconds for detecting one sample image and 0.082 seconds for detecting speech in a second of voice. The findings and experiments confirm the applicability and effectiveness of the proposed approach for real-time cheating detection in the online exam. For future work, we will perform an empirical study on large recorded datasets and parameters with different types of cheating behaviors.

CRediT authorship contribution statement

Sanaa Kaddoura: Conceptualization, Formal analysis, Methodology, Writing – original draft, Writing – review & editing. **Abdu Gumaei:** Conceptualization, Formal analysis, Methodology, Writing – original draft, Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

The data is publicly available online.

Acknowledgement

This work is funded by Zayed University Research Incentive Fund (RIF) with grant number R20128.

References

Atoum, Y., Chen, L., Liu, A. X., Hsu, S. D., & Liu, X. (2017). Automated online exam proctoring. *IEEE Transactions on Multimedia*, *19*, 1609–1624.

Cappé, O. (1994). Elimination of the musical noise phenomenon with the Ephraim and Malah noise suppressor. *IEEE Transactions on Speech and Audio Processing*, *2*, 345–349.

Chaudhari, P., Agrawal, H., & Kotecha, K. (2020). Data augmentation using MG-GAN for improved cancer classification on gene expression data. *Soft Computing*, *24*, 11381–11391.

Draaijer, S., Jefferies, A., & Somers, G. (2017). Online proctoring for remote examination: A state of play in higher education in the EU. In *International conference on technology enhanced assessment* (pp. 96–108). Springer.

Ephraim, Y., & Malah, D. (1984). Speech enhancement using a minimum-mean square error short-time spectral amplitude estimator. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, *32*, 1109–1121.

Gopane, S., & Kotecha, R. (2022). Enhancing monitoring in online exams using artificial intelligence. In *Proceedings of international conference on data science and applications* (pp. 183–193). Springer.

Gumaei, A., Al-Rakhami, M., Hassan, M. M., Alamri, A., Alhussein, M., Razzaque, M., Fortino, G., et al. (2020). A deep learning-based driver distraction identification framework over edge cloud. *Neural Computing and Applications*, 1–16.

Gumaei, A., Ismail, W. N., Hassan, M. R., Hassan, M. M., Mohamed, E., Alelaiwi, A., & Fortino, G. (2022). A decision-level fusion method for Covid-19 patient health prediction. *Big Data Research*, *27*, Article 100287.

Hussein, M. J., Yusuf, J., Deb, A. S., Fong, L., & Naidu, S. (2020). An evaluation of online proctoring tools. *Open Praxis*, *12*, 509–525.

Jalali, K., & Noorbehbahani, F. (2017). An automatic method for cheating detection in online exams by processing the students webcam images. In *3rd conference on electrical and computer engineering technology (E-Tech 2017)* (pp. 1–6).

Joshi, G., Walambe, R., & Kotecha, K. (2021). A review on explainability in multimodal deep neural nets. *IEEE Access*, *9*, 59800–59821.

Kadam, K., Ahirrao, S., Kotecha, K., & Sahu, S. (2021). Detection and localization of multiple image splicing using MobileNet v1. *IEEE Access*, *9*, 162499–162519.

Kaddoura, S., Popescu, D. E., & Hemanth, J. D. (2022). A systematic review on machine learning models for online learning and examination systems. *PeerJ Computer Science*, *8*, Article e986.

Karlos, S., Kostopoulos, G., & Kotsiantis, S. (2020). A soft-voting ensemble based co-training scheme using static selection for binary classification problems. *Algorithms*, *13*, 26.

King, D. L., & Case, C. J. (2014). E-cheating: Incidence and trends among college students. *Issues in Information Systems*, *15*.

Lee, K., & Fanguy, M. (2022). Online exam proctoring technologies: Educational innovation or deterioration? *British Journal of Educational Technology*.

Lefter, I., Rothkrantz, L. J., & Burghouts, G. J. (2013). A comparative study on automatic audio–visual fusion for aggression detection using meta-information. *Pattern Recognition Letters*, *34*, 1953–1963.

Li, X., Chang, K.-m., Yuan, Y., & Hauptmann, A. (2015). Massive open online proctor: Protecting the credibility of MOOCs certificates. In *Proceedings of the 18th ACM conference on computer supported cooperative work & social computing* (pp. 1129–1137).

Liu, Z.-Y., Lomovtseva, N., & Korobeynikova, E. (2020). Online learning platforms: Reconstructing modern higher education. *International Journal of Emerging Technologies in Learning*, *15*, 4–21.

Malhotra, N., Suri, R., Verma, P., & Kumar, R. (2022). Smart artificial intelligence based online proctoring system. In *2022 IEEE Delhi section conference (DELCON)* (pp. 1–5). IEEE.

Masud, M. M., Hayawi, K., Mathew, S. S., Michael, T., & El Barachi, M. (2022). Smart online exam proctoring assist for cheating detection. In *International conference on advanced data mining and applications* (pp. 118–132). Springer.

Milone, A. S., Cortese, A. M., Balestrieri, R. L., & Pittenger, A. L. (2017). The impact of proctored online exams on the educational experience. *Currents in Pharmacy Teaching and Learning*, *9*, 108–114.

Nguyen, L. S., Frauendorfer, D., Mast, M. S., & Gatica-Perez, D. (2014). Hire me: Computational inference of hirability in employment interviews based on nonverbal behavior. *IEEE Transactions on Multimedia*, *16*, 1018–1031.

Nigam, A., Pasricha, R., Singh, T., & Churi, P. (2021). A systematic review on AI-based proctoring systems: Past, present and future. *Education and Information Technologies*, *26*, 6421–6445.

Noorbehbahani, F., Mohammadi, A., & Aminazadeh, M. (2022). A systematic review of research on cheating in online exams from 2010 to 2021. *Education and Information Technologies*, 1–48.

O'Reilly, G., & Creagh, J. (2016). A categorization of online proctoring. In *Global learn* (pp. 542–552). Association for the Advancement of Computing in Education (AACE).

Prathish, S., Bijlani, K., et al. (2016). An intelligent system for online exam monitoring. In *2016 international conference on information science (ICIS)* (pp. 138–143). IEEE.

Reale, M. J., Canavan, S., Yin, L., Hu, K., & Hung, T. (2011). A multi-gesture interaction system using a 3-d iris disk model for gaze estimation and an active appearance model for 3-d hand pointing. *IEEE Transactions on Multimedia*, *13*, 474–486.

Rosen, W. A., & Carr, M. E. (2013). An autonomous articulating desktop robot for proctoring remote online examinations. In *2013 IEEE frontiers in education conference (FIE)* (pp. 1935–1939). IEEE.

Sapre, S., Shinde, K., Shetta, K., & Badgujar, V. (2022). AI-ML based smart online examination framework. In *International conference on deep learning, artificial intelligence and robotics* (pp. 17–25). Springer.

Slusky, L. (2020). Cybersecurity of online proctoring systems. *Journal of International Technology and Information Management*, *29*, 56–83.

Sohn, J., Kim, N. S., & Sung, W. (1999). A statistical model-based voice activity detection. *IEEE Signal Processing Letters*, *6*, 1–3.

Sohn, J., & Sung, W. (1998). A voice activity detector employing soft decision based noise spectrum adaptation. In *Proceedings of the 1998 IEEE international conference on acoustics, speech and signal processing, ICASSP'98 (Cat. No. 98CH36181), vol. 1* (pp. 365–368). IEEE.

Wahid, A., Sengoku, Y., & Mambo, M. (2015). Toward constructing a secure online examination system. In *Proceedings of the 9th international conference on ubiquitous information management and communication* (pp. 1–8).

Walambe, R., Marathe, A., Kotecha, K., & Ghinea, G. (2021). Lightweight object detection ensemble framework for autonomous vehicles in challenging weather conditions. *Computational Intelligence and Neuroscience*, 2021.

Walambe, R., Nayak, P., Bhardwaj, A., & Kotecha, K. (2021). Employing multimodal machine learning for stress detection. *Journal of Healthcare Engineering*, *2021*.

Xiao, B., Georgiou, P., Baucom, B., & Narayanan, S. S. (2015). Head motion modeling for human behavior analysis in dyadic interaction. *IEEE Transactions on Multimedia*, *17*, 1107–1119.

Dr. Sanaa Kaddoura holds a Ph.D. in computer science from Beirut Arab University, Lebanon. She is currently employed as an assistant professor of information security at the Department of Computing and Applied Technology, College of Technological Innovation, Zayed University, United Arab Emirates. She is also an assistant professor of business analytics for master's degree students in the UAE. She is a fellow of Higher Education Academy, Advance HE (FHEA) since 2019, which demonstrates a personal and institutional commitment to professionalism in learning and teaching in higher education. Furthermore, she is a certified associate from Blackboard academy since April 2021. In addition to her research interests in cybersecurity, social networks, machine learning, and natural language processing, she is an active researcher in higher education teaching and learning related to enhancing the quality of instructional delivery to facilitate students' acquirement of skills and smooth transition to the workplace.

Dr. Abdu Gumaei received his Ph.D. degree in Computer Science from King Saud University in 2019. He is currently an Assistant Professor at Computer Science Department, Taiz University, Yemen. He worked as a lecturer and taught many courses such as programming languages in the computer science department, Taiz University, Yemen. He has authored more than 90 journal and conference papers in well-reputed international journals. He has received a patent from the UNITED STATES PATENT AND TRADEMARK OFFICE (USPTO) in 2013. His main areas of interest are software engineering, image processing, computer vision, machine learning, networks, and the Internet of Things (IoT).