Enhancing convolutional neural network based model for cheating at online examinations detection

Sara Ouahabi[1], Rihab Aboudihaj[1], Nawal Sael[1], Kamal El Guemmat[2]
[1]Laboratory of Modeling and Information Processing, Faculty of Sciences Ben M'sik, Hassan II University, Casablanca, Morocco
[2]Laboratory of Signals, Distributed Systems, and Artificial Intelligence, ENSET, Hassan II University, Mohammedia, Morocco

Article Info

Article history:

Received Mar 18, 2024
Revised Nov 6, 2024
Accepted Nov 14, 2024

Keywords:

Cheating detection
Convolutional neural network
Deep learning
Head mouvement analysis
Online exams

ABSTRACT

In the last few years, e-learning has revolutioning education, giving students access to diverse and adaptable on-line resources, but it has also face a major challenge: cheating on online exams. Students now use variant cheating methods include consulting unauthorized documents, communicating with others during the exam, searching for information on the internet. Combating these cheating practices has become crucial to preserving the integrity of academic assessments. In this context, artificial intelligence (AI) has emerged as an essential tool for mitigating this fraudulent behavior. Equipped with advanced machine learning capabilities, AI can examine a wide range of data to detect student suspicious behavior. This study develops an approach based on a convolutional neural network (CNN) model designed to detect cheating by analyzing candidates' head movements during online exams. By exploiting the FEI dataset, this model achieves an interesting accuracy of 97.28%. In addition, we compare this model to the well-known transfer learning models used in the literature namely, ResNet50, VGG16, DenseNet21, MobileNetV2, and EfficientNetB0 demonstrating the out performance of our approach in detecting cheating during online exams.

Corresponding Author:

Sara Ouahabi
Laboratory of Modeling and Information Processing, Faculty of Sciences Ben M'sik, Hassan II University
Casablanca, Morocco
Email: s3.ouahabi@gmail.com

1. INTRODUCTION

E-learning is playing a vital role in the existing educational setting, as it changes the entire education system and becomes one of the greatest preferred topics for academics [1]. This shift has been driven by the need for a safe and efficient alternative to in-person learning. In fact, e-learning allow providing effective teaching methods, catering to diverse learning styles and offers accessibility to a vast array of educational resources and interactive opportunities [2], [3], promoting active engagement and critical thinking. However, e-learning faces also major challenges such as cheating. Exam fraud is widespread globally [4]–[6], whatever the level of development. As a result, traditional cheating detection methods may no longer be totally effective in preventing examination fraud. Online exams are an integral part of e-learning solutions for authentic and fair assessment of student performance [7]. The design and execution of online exams are the most challenging aspects of e-learning. In particular, online exams are usually conducted on e-learning platforms without the physical presence of students and instructors in the same place. This creates several deficiencies in terms of the integrity and security of online exams. For example, candidate authenticity verification is extremely problematic in an online environment, particularly in the absence of continuous monitoring. What's more, online exams are highly conducive to cheating, as thousands of information resources are accessible to students without any controls. In this context, where preserving the integrity of online exams is crucial, artificial intelligence (AI) offers advanced analysis and detection skills, making it an invaluable asset for guaranteeing the reliability of online assessments. Cheating on online exams can be detected and classified in multitude forms, from collusion between students to the use of mobile devices, such as phones, to more subtle indicators such as eye movements [8], [9], mouth movements head movements [10], and many other unauthorized behaviors.

In the face of the diverse methods of cheating, this study concentrates on analyzing candidates' head movements during exams since it is considered as key element to detect several other behaviors and serve as potentially significant indicators of cheating. The primary goal is to employ the convolutional neural network (CNN) approach on the FEI dataset, specifically designed to detect head movements. In addition, a comparison of the approach's performance against other models including VGG16, ResNet50, DenseNet21, EfficientNetB0, and MobileNetV2 is performed to evaluate its efficacy in cheating detection is done. Our approach outperforms the other models and has the potential to significantly contribute to the prevention of cheating in online exams by offering a precise and dependable solution to this intricate issue.

The remainder of this article is structured as follow. Section 2 provides an in-depth review of related work in the field of online cheating detection. Section 3 details the methodology developed, which is centered on deep learning (DL) for analyzing candidates' head movements. In section 4, more details about the experiment and the obtained results are presented, offering essential insights into the approach's effectiveness. Finally, section 5 serves as the conclusion of the study.

2. RELATED WORKS

In the field of online assessment, which is evolving rapidly, researchers face several challenges and explore numerous research possibilities. Many studies have been conducted to improve the integrity of online exams and address the issue of cheating. Bawarith *et al.* [11] proposes a methodology based on continuous authentication, eye tracking, and fingerprint scanning, which was applied to a private dataset. The results demonstrated a sensitivity of 100%, a specificity of 95.56%, an accuracy of 95.74%, an overall accuracy of 97.78%, and an F-measure of 97.83. Furthermore, Jalali and Noorbehbahani [12] presents two distinct methods for detecting cheating in online exams, both of them were developed and tested using a private dataset. The first method is based on image processing using MATLAB, calculating the difference between pixels in the images. When the threshold is set at 9, this method boasts high accuracy, particularly for detecting empty seats, with an accuracy of 100%, however, it is sensitive to variations in the color of walls, students' clothing and objects such as sheets of paper. The second method is based on clustering reference images using the k-medoids algorithm. This method reduces complexity compared to the first method, but its accuracy is slightly lower, with an average accuracy of 68%. It excels in detecting empty seats, also achieving 100% accuracy. Geetha *et al.* [13] used the Eigenface method for extracting facial features from facial vectors and the support vector machine (SVM) model to improve detection accuracy. This approach was applied to a private dataset, and the obtained matching accuracy was approximately 61% with 50 real-time images in the dataset. This accuracy can be improved by increasing the number of images in the dataset.

To addresses the issue of cheating in exams, whether in paper or electronic exam copies, Rhein *et al.* [14] introduce FLEXauth, an application that utilizes AI techniques for author verification in electronic programming exams using a private dataset. The idea is based on the principle that each student develops an individual style for answering certain types of exercises, which can be extracted using AI tools and then compared to reference material with verified authors. The concept is applied to Java programming exams, but the goal is to extend support to other programming languages and types of assignments in the future. The results show that the random forest (RF) method achieved the best performance with an accuracy of up to 67.15% for classifying the top three options. Moreover, the GoogleCodeJam2017 dataset was used for testing, where an accuracy of 93.80% was achieved.

In addition to machine learning (ML) and authentification based cheating detection techniques, others papers used DL techniques and transfer learning. In fact, Ozdamli *et al.* [15] employed computer vision algorithms from the OpenCV library for image acquisition, preprocessing, feature extraction, detection/segmentation, high-level processing, and decision making. CNNs, particularly the mini Xception model, were used to predict facial emotions. Various datasets were used for training and testing, including images for facial verification, emotion recognition, gaze tracking, and head movements. The system achieved an accuracy of approximately 99.38% on the LFW dataset. For in-class emotion tracking, the average accuracy was approximately 66% on the FER dataset. Finally, for monitoring behaviors during online exams, the system displayed an eye-tracking accuracy of approximately 96.95% on the gi4E dataset and a head movement tracking accuracy of approximately 96.24% on the FEI dataset. Yulita *et al.* [16] use MobileNetV2 architecture for recognizing activities during online exams, which was applied to the OEP dataset. Optimal hyperparameters were found, resulting in an F1 score of 84.52%. A web application was developed to assist teachers in detecting academic fraud. This represents a significant advancement in educational technology.

In various studies [17]–[20], researchers have explored emotion recognition from facial expressions from different perspectives. El Hammoumi *et al.* [17] focused on developing a facial expression recognition system based on CNNs, with a clear goal of integration into an e-learning system. This methodology involved image preprocessing, feature extraction, and image classification, resulting in an impressive test accuracy of 97.53% on CK+ and 97.18% on Japanese female facial expression (JAFFE). Ozdemir *et al.* [18] adopted a similar approach using DL and CNNs to create an emotional classification model, merging various datasets. Their model based on the LeNet architecture achieved a training accuracy of 96.43% and a validation accuracy of 91.81%. Furthermore, Pranav *et al.* [19] focused on developing a deep convolutional neural network (DCNN) model to classify five different human facial emotions, achieving an accuracy of 78.04%. Finally, Zbaida *et al.* [20] examined the importance of emotion recognition in human-machine interaction, especially in e-learning platforms, by evaluating various approaches such as VGG16, VGG19, ResNet50V2, EfficientNetB0, and EfficientNetB7 to develop a specialized neural network for accurate emotion identification.

The analysis of the current state of the art, reveals that CNNs have become important for their excellent performance in various online assessment areas. Many different types of CNNs have been studied in different situations and with different data, showing flexibility and potential. The related works studies have utilized CNNs for various tasks such as recognizing emotions, analyzing facial expressions, and monitoring behavior during online exams. However, when exploring research, we can conclude that great efforts are still needed to enhance the performance of CNNs, especially in detecting cheating. The focus of this research is on improving CNNs using the FEI dataset designed specifically for detecting head movements.

The choice to focus on the analysis of head movements was made due to its significance in comprehending a student's behavior during an examination. Detecting head movements is a pivotal approach for recognizing potential student cheating, and it constitutes a fundamental aspect of the research. An overview of the various studies covered in the state of the art is provided. This summary aims to provide a concise perspective on the objectives, datasets, methodologies used and results obtained by each article included in this literature review. Furthermore, in the results column, the abbreviation "Acc" stands for "Accuracy", representing the accuracy value obtained by each method adopted, as detailed in Table 1.

Table 1. Summary table of reviewed articles

Ref	Objective	Dataset name	Dataset size	Methods	Results (%)
[11]	Eye movement detection	Private	30 students	Fingerprint, Eye tracking, E-proctor	Acc=97.78
[12]	Behavior detection	Private	50 images of students	MATLAB, K-Medoids	Acc=78
[13]	Facial recognition	Private	50 images of students	Face detector based on caffe, Eigenface, SVM	Acc=61
[14]	Analyzing students responses	Private	2 assignments from 13 students	DNN	Acc=37.78
		Google CodeJam2017	40 participants	RF	Acc=67.15
				SVM	Acc=57.69
[15]	Emotion recognition	FER	35600 images	CNN, MiniXception	Acc=66
	Facial recognition	LFW	13250 images of 5,794 individuals	Deep metric learning, ResNet34	Acc=99.38
	Eye tracking	Gi4E	1380 images	Trained model based on eye tracking	Acc=96.95
	Head movements detection	FEI	2800 images of 200 individuals	Haar cascade	Acc=96.24
	Object detection	COCO	5000 images	YOLO (V3)	Acc=60
[16]	Behavior detection	OEP	24 participants	MobileNetV2	Acc=86.7
[17]	Emotion detection	CK+	327 sequences of images	CNN, Haar cascade	Acc=97.53
		KDEF	4900 face images from 70 subjects		
		JAFFE	213 images of 7 Japanese women		Acc=97.18
[18]	Emotion detection	Private	140 images with 7 expressions	LeNet CNN architecture, Haar cascade	Acc=96.43
		KDEF	4900 face images from 70 subjects		
		JAFFE	213 images of 7 Japanese women		
[19]	Emotion detection	Private	2550 images of 5 emotions	DCNN	Acc=78.04
[20]	Emotion recognition	FER	35600 images	VGG16, VGG19, ResNet50, EfficientNetB0, EfficientNetB7	Acc=98.45

3. METHOD

The main objective of this paper is to implement a CNN model architecture for head movement detection, with the aim of identifying cheating behavior in online exams using the FEI dataset. In this section, the methodology used is detailed, and the crucial steps followed to ensure the rigor and reliability of the results are highlighted. The approach is structured in several fundamental steps. First, a data collection was carrifed out. This initial phase was followed by a critical data cleaning step aimed at eliminating any inconsistencies or anomalies. Next, the data was labeled into two distinct categories: cheating and non-cheating, to facilitate the identification of behavioral patterns. Afterwards, a data pre-processing phase was carried out. This included data augmentation to diversify and enrich the dataset, as well as images resizing to ensure consistency of model inputs. To maintain a balance between classes, a balancing strategy was implemented. Finally, DL models were implemented and their performance was evaluated in the context of the study. Figure 1 presents an illustrative diagram of the methodology adopted.

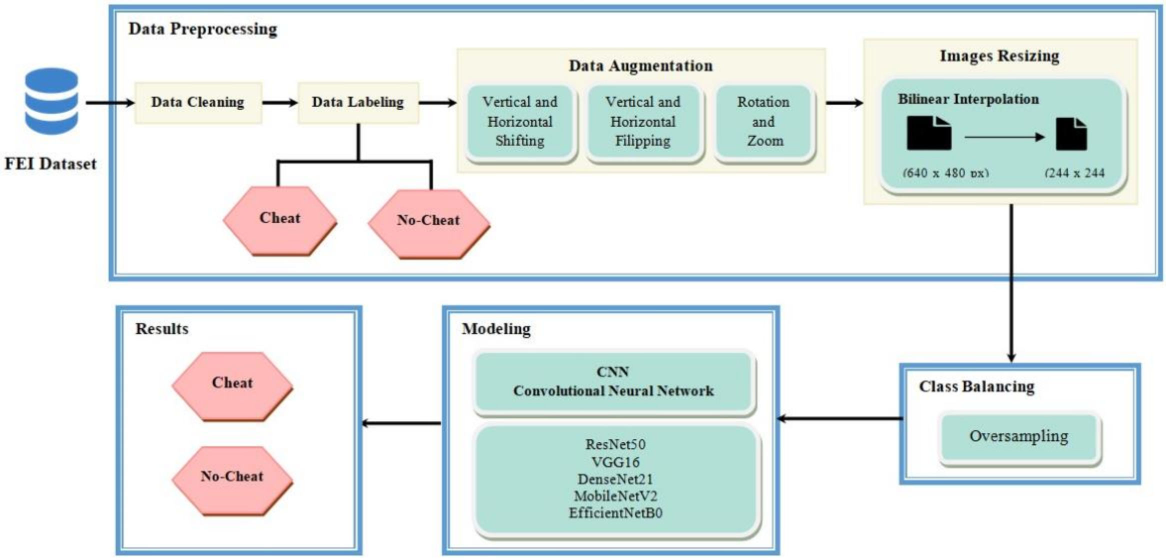

Figure 1. The proposed approach

3.1. Data collection

In this study, the FEI dataset was used. This dataset was gathered between June 2005 and March 2006 at the FEI AI Laboratory in São Bernardo do Campo, São Paulo, Brazil. It comprises 14 images per individual from among 200 distinct individuals, totaling 2800 images. Each of these images is in color, captured in a straight frontal position on a uniform white background, with profile rotations of up to approximately 180 degrees. Several features of this dataset make it particularly suitable for the task. Firstly, it provides a variety of poses, covering an impressive range of profile rotations, including lateral head movements. This diversity is crucial for the project as it reflects the different head positions aimed to detect in cases of cheating. Additionally, the consistent size of images in the FEI dataset is essential for ensuring the consistency of visual features that DL models can learn. This consistency simplifies the training of models. Furthermore, it is gender-balanced, with an equal number of male and female subjects (100 of each). This gender balance is essential to avoid any gender bias in the detection model.

3.2. Data cleaning

The FEI dataset, while rich in head movement information, contains both images of human faces and embedded images without faces. To ensure the quality of our training dataset, we undertook a cleaning process. In this phase, we carefully examined every image in the FEI dataset and eliminated those that did not contain human faces. This was essential to avoid any potential confusion of our model during training, ensuring that only images relevant to head movement detection were retained. The cleaning process was carried out with great attention to detail, and each image was assessed for its suitability to our task.

3.3. Data labeling

The dataset used in the study was labeled into two distinct classes: the "non-cheating" class and the "cheating" class. The "non-cheating" class includes images of individuals concentrating on the screen during an online exam. On the other hand, the "cheating" class includes images of individuals positioned outside the screen field, turning their heads to the right or left, or adopting behaviors suggestive of cheating. This labeling step enabled the researchers to specifically target the cheating behaviors they were seeking to detect.

3.4. Data augmentation

The dataset used in the study was relatively small, so data augmentation was applied to increase its diversity and size. New images were generated from existing ones by applying various transformations, including vertical and horizontal shifting, horizontal and vertical flipping, rotation, and zooming. These techniques were applied in a controlled manner to create variations of the original images, increasing the amount of data while maintaining relevance to the detection task. Data augmentation is an essential tool for reinforcing the robustness of the model by exposing it to a wider variety of possible situations while avoiding overfitting. To provide a clear perspective on the effect of data augmentation, Table 2 shows the number of images in the dataset before and after augmentation, respectively. This table highlights the extent to which the dataset has been enriched by this technique.

Table 2. Number of images before and after applying data augmentation

Class	Number of images before applying data augmentation	Number of images after applying data augmentation
Cheat	1479	4437
No-cheat	1186	3558

3.5. Images resizing

Another essential step in the data preparation process involved image resizing. The original dataset included images with a resolution of 640×480 pixels, which was unsuitable for the model. Therefore, all images were resized to a uniform resolution of 224×224 pixels. This resizing was done in such a way as to retain the essential visual information while reducing the complexity of the data, which is beneficial for training DL models. This step ensures that the input data is uniform and ready to be processed by the cheat detection algorithms while reducing the computational load required for image processing.

3.6. Class balancing

To address the marked imbalance between the cheating and non-cheating classes, the oversampling technique was implemented to manage class imbalance. This method consists of increasing the number of examples of the minority class by duplicating or generating new instances. It can be useful when there are a small number of examples of the minority class and the goal is to balance the classes. The related works provided support the use of oversampling as a technique to address class imbalance in ML, particularly in the context of cheating detection in large-scale assessments. This class imbalance management is an important step in ensuring the accuracy and reliability of the cheating detection model by minimizing the risk of false positives or false negatives linked to the initial imbalance of the data.

3.7. Modeling

In this study, the cheating detection model is based on CNN architecture. However, in order to evaluate the effectiveness of the approach, several other reference models were also included in the analysis. These reference models are identified from the literature review and derived from different deep neural network and serve as points of comparison to assess the capability of the applied CNN model. In the following section, a closer look is taken at these reference models.

3.7.1. Convolutional neural network

CNN is a network model proposed by Lecun [21] are a class of neural networks that have proved highly effective in fields such as image recognition and classification. CNNs are a type of forward-propagating neural network composed of several layers. They consist of filters, kernels or neurons with adjustable weights or parameters and biases. Each filter takes inputs, performs a convolution, and eventually applies a non-linearity. The structure of a CNN includes convolution, pooling, rectified linear unit (ReLU) and fully connected layers [22]. To evaluate the outperformance of our proposed CNN model, we have explored the most performant DL models used in the literature review. The models adopted are as follows.

3.7.2. Residual network 50

In principle, adding extra layers to a neural network should improve model quality provided the overfitting problem is addressed. However, much architecture face the challenge of "vanishing gradients". The ResNet architecture has been designed to solve this problem by introducing shortcut connections. These connections ensure that adding layers does not require learning an identical transformation to maintain or surpass the performance of architecture with fewer layers. This is made possible by the immediate addition of a direct connection between the output of each layer and the input of the next layer [23].

3.7.3. Visual geometry group-16

Is a CNN architecture extensively employed in diverse computer vision tasks, particularly in emotion recognition based on facial expressions. Developed by the VGG at the University of Oxford, the VGG16 model stands out for its deep structure, comprising 16 layers, including 13 convolutional layers and 3 fully connected layers. Notable for its simplicity and consistency, the VGG16 architecture features convolutional layers with a small receptive field (3×3), stacked sequentially. This architectural approach enables the model to acquire intricate features by capturing local patterns in the early layers, progressively moving on to more complex patterns as the depth increases [24].

3.7.4. Densely connected convolutional networks 21

Represents a contemporary CNN architecture designed for visual object recognition, achieving state-of-the-art performance with a reduced number of parameters. While sharing some fundamental similarities with ResNet, DenseNet introduces notable modifications. Unlike ResNet's additive attribute (+) that merges previous and future layers, DenseNet employs concatenation (.) to combine the output of the previous layer with that of the subsequent layer. This architectural distinction addresses the connectivity challenge by densely interconnecting all layers [25].

3.7.5. MobileNetV2

Is a compact, swift, and precise DCNN tailored for classification and detection assignments. Engineered to excel in terms of both speed and size efficiency. This network ensures noteworthy accuracy in various computer vision tasks such as object classification and detection [26].

3.7.6. EfficientNet

Stands out as a CNN architecture acclaimed for its efficiency and outstanding performance across diverse computer vision tasks, including the discernment of emotions based on facial expressions. Employing a compound scaling approach, the EfficientNetB0 architecture uniformly adjusts the network's depth, width, and resolution. This scaling strategy enables the model to strike a commendable balance between capacity and computational efficiency [27].

4. EXPERIMENTS AND RESULTS

In this section, we present the experiments conducted and the corresponding results obtained. The primary aim of these experiments was to assess the effectiveness of the proposed CNN model architecture in detecting head movements during online exams. We also compare its performance with other models to evaluate its capability.

4.1. Proposed convolutional neural network

The CNN model follows a multi-stage architecture. Firstly, the first phase of the model comprises three convolution layers, with a ReLU activation function, which take as input images of size 244×244×3. After each convolution, a max-pooling layer with a pool window of 2×2 is applied to reduce the dimensions. A dropout layer is then inserted at a rate of 0.5 to reduce overfitting. The second phase begins with a flatten layer that transforms the two-dimensional data into a one-dimensional array. Next, two fully connected layers (dense) with a ReLU activation function are added to learn more abstract features in the data. Finally, for the output of the model, a dense layer with a Sigmoid activation function is included, indicating a binary classification (cheat or non-cheat). The model is compiled with the Adam optimizer, a binary_crossentropy loss function adapted to binary classification. Figure 2 shows a summary of the CNN architecture adopted. To enhance the performance, speed and relevance of our proposed DL models, we have adopted the following hyperparameters, as presented in Table 3. These hyperparameters were carefully selected to optimize training efficiency and ensure that the model converges appropriately.

4.2. Results of the proposed convolutional neural network model

After successfully training the CNN model, an accuracy of 97.28% was achieved. For detailed results, including the evolution of accuracy and loss over the 20 training epochs, refer to Figures 3 and 4. These graphs provide a clear representation of the model's performance over time. Based on Figure 3, the evolution of the CNN model over 20 training epochs indicates a continuous improvement in its performance. The accuracy on the validation set gradually increases, showcasing the model's enhanced predictive capabilities. This consistent upward trajectory highlights the effectiveness of the training process and reflects the model's growing ability to generalize to unseen data. In contrast, Figure 4 illustrates the evolution of loss over the same 20 training epochs. Remarkably, the loss decreases significantly, reflecting the model's ability to optimize its predictions. These findings collectively demonstrate the positive progression and effectiveness of the CNN model throughout the training process. Furthermore, the reduction in loss corroborates the model's capability to minimize errors, thereby enhancing its overall performance.

```
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d_7 (Conv2D)           (None, 242, 242, 32)      896

 max_pooling2d_7 (MaxPoolin  (None, 121, 121, 32)      0
 g2D)

 conv2d_8 (Conv2D)           (None, 119, 119, 64)      18496

 max_pooling2d_8 (MaxPoolin  (None, 59, 59, 64)        0
 g2D)

 conv2d_9 (Conv2D)           (None, 57, 57, 128)       73856

 max_pooling2d_9 (MaxPoolin  (None, 28, 28, 128)       0
 g2D)

 flatten_2 (Flatten)         (None, 100352)            0

 dense_4 (Dense)             (None, 256)               25690368

 dropout_2 (Dropout)         (None, 256)               0

 dense_5 (Dense)             (None, 1)                 257

=================================================================
Total params: 25783873 (98.36 MB)
Trainable params: 25783873 (98.36 MB)
Non-trainable params: 0 (0.00 Byte)
```

Figure 2. CNN model architecture used

Table 3. Hyperparameters

Category	Hyperparameters	Value or configuration
Layer hyperparameters	Dropout	50%
	Kernel size	(3x3)
	Activation function of the final layer	Sigmoid
	Activation function of hidden layers	ReLU
Compilation hyperparameters	Optimization function	Adam
	Error function	binary_crossentropy
	Learning rate	0.001
Execution hyperparameters	Batch size	32
	Number of epochs	20

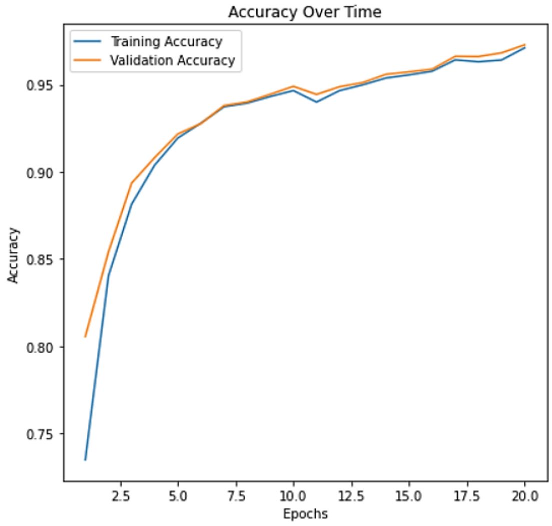

Figure 3. Evolution of training and validation accuracy

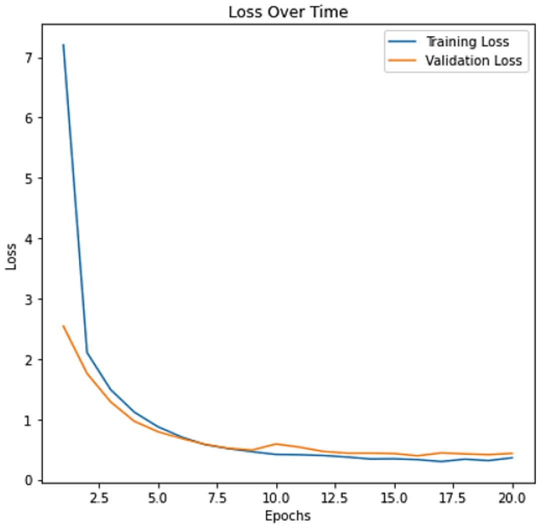

Figure 4. Evolution of of training and validation loss

Additionally, Figure 5 presents the confusion matrix, allowing for a comprehensive assessment of the model's classification capability. This matrix provides valuable insights into the model's strengths and weaknesses, highlighting areas for potential improvement. The confusion matrix obtained after training the model to classify cheating and non-cheating cases reveals impressive performance, meaning that the model correctly identified cheating and non-cheating cases in the majority of situations.

After examining the obtained metrics, it is clear that the CNN model has achieved outstanding performance. With an accuracy of 97.28%, the model has demonstrated a strong ability to correctly classify input data. Furthermore, a precision score of 98.00% attests to its capability in minimizing false positives. These results reveal the efficiency and reliability of the model, positioning it as a promising solution for the given task. They also confirm that the model is well-tailored to the data and provides a suitable balance between identifying true positives and minimizing classification errors. These results are encouraging for the future of the project and suggest that the model is ready for deployment in real-world scenarios.

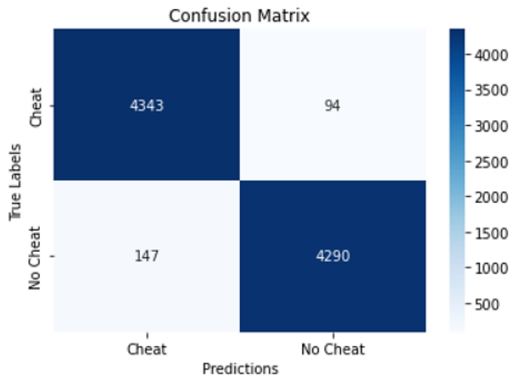

Figure 5. Confusion matrix

4.3. Comparison with convolutional neural network-based pre-trained models

In the evaluation phase, the aim is to assess the performance of the CNN model by comparing it with several widely recognized pre-trained architectures, including ResNet50, VGG16, DenseNet121, MobileNetV2, and EfficientNet. Once various models were successfully trained, a detailed comparison of the results obtained with the CNN model was conducted. In order to summarize these results and facilitate the comparison, consult Table 4 for a summary of all the metrics and performances recorded. This comparative assessment will enable a better evaluation of the CNN model in use.

After a series of in-depth training sessions and evaluations, we are able to observe the performance of each model. These training and evaluation stages were crucial to understanding how each model performed in the specific context of our online exam cheating detection project. The final results reveal that, among the models evaluated, our CNN architecture achieved the most performance, with an accuracy of 97.28%. Compared to other commonly used architectures, such as ResNet, MobileNetV2, EfficientNet, DenseNet, and VGG16, our CNN model stands out significantly. For example, although ResNet is recognized as a leading architecture, our CNN model outperforms it with an accuracy of 97.28%, compared with 95.24% for ResNet. Similarly, MobileNet achieved 96.93% accuracy.

Table 4. Models metrics

Models	Accuracy	Precision	Recall	F1-score
Proposed CNN model	97.28	98.00	96.60	97.30
ResNet50	95.24	97.00	92.00	95.10
MobileNetV2	96.93	95.20	98.00	97.00
DenseNet21	94.88	96.80	92.80	94.70
VGG16	96.32	94.10	98.70	96.40
EfficientNetB0	96.12	97.00	95.80	96.40

On the other hand, the 97.28% accuracy rate achieved illustrates the remarkable effectiveness of the method in classifying head movements in the FEI dataset, which is divided into two distinct classes: cheating and non-cheating. Comparing these results with those of previous studies documented in the literature, also conducted on the same FEI dataset, it is interesting to note that these earlier works achieved an accuracy rate of 96.24% [15]. Therefore, these results significantly surpass previously reported performance, demonstrating that the CNN-based approach represents a substantial advance in solving this particular problem.

5. CONCLUSION

In conclusion, this research explored the use of DL to detect cheating in online exams, focusing specifically on the detection of head movements. The proposed CNN model demonstrated exceptional performance, outperforming other evaluated models such as ResNet, VGG16, DenseNet21, MobileNetV2, and EfficientNetB0. The proposed approach achieved 97.28% accuracy, with 98.00% precision and 96.60% recall. These results are significantly better than previous work on the same FEI dataset, which achieved a precision of 96.24%. The effectiveness of the CNN model in detecting cheating in online exams suggests that it could be successfully deployed in real-world scenarios, helping to ensure the integrity of online assessments. However, it is important to note that this model is not infallible and may not be as effective in detecting more subtle forms of cheating, such as verbal cheating or discreet collaboration between students. These forms of cheating may not involve visible head movements, making their detection more difficult for our current model.

6. PERSPECTIVES

In our future work, we plan to explore other perspectives and study various forms of online cheating, including those that are more complex and difficult to detect. For example, we plan to study eye movements, facial expressions and voice recognition as potential means of detecting online cheating. By expanding our cheating detection toolkit, we hope to strengthen our model and improve its ability to detect a wider range of online cheating behaviors.

REFERENCES

[1] A. M. Maatuk, E. K. Elberkawi, S. Aljawarneh, H. Rashaideh, and H. Alharbi, “The covid-19 pandemic and e-learning: challenges and opportunities from the perspective of students and instructors,” *Journal of Computing in Higher Education*, vol. 34, no. 1, pp. 21–38, 2022, doi: 10.1007/s12528-021-09274-2.

[2] S. Ouahabi, A. Eddaoui, E. H. Labriji, E. Benlahmar, and K. El Guemmat, “Implementation of a novel educational modeling approach for cloud computing,” *International Journal of Emerging Technologies in Learning*, vol. 9, no. 6, pp. 49–53, 2014, doi: 10.3991/ijet.v9i6.4153.

[3] S. Ouahabi, K. El Guemmat, M. Azouazi, and S. El Filali, “A survey of distance learning in morocco during covid-19,” *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 22, no. 2, pp. 1087–1095, 2021, doi: 10.11591/ijeecs.v22.i2.pp1087-1095.

[4] A. A. Bamanikar, A. Patil, D. Singh, M. Kumbhar, and V. Badhe, “Online exam portal using machine learning and face detection,” *International Journal for Research in Applied Science and Engineering Technology*, vol. 11, no. 5, pp. 4624–4630, 2023, doi: 10.22214/ijraset.2023.52692.

[5] A. M. Sendur, “Academic malpractice in tests and exams from an international perspective,” *Przegląd Badań Edukacyjnych*, vol. 36, no. 1, 2022, doi: 10.12775/pbe.2022.007.

[6] L. C. O. Tiong and H. J. Lee, “E-cheating prevention measures: detection of cheating at online examinations using deep learning approach -- a case study,” *arXiv-Computer Science*, pp. 1–9, 2021.

[7] A. W. Muzaffar, M. Tahir, M. W. Anwar, Q. Chaudry, S. R. Mir, and Y. Rasheed, “A systematic review of online exams solutions in e-learning: techniques, tools, and global adoption,” *IEEE Access*, vol. 9, pp. 32689–32712, 2021, doi: 10.1109/ACCESS.2021.3060192.

[8] N. Dilini, A. Senaratne, T. Yasarathna, N. Warnajith, and L. Seneviratne, “Cheating detection in browser-based online exams through eye gaze tracking,” in *2021 6th International Conference on Information Technology Research (ICITR)*, IEEE, 2021, pp. 1–8, doi: 10.1109/ICITR54349.2021.9657277.

[9] A. Javed and Z. Aslam, “An intelligent alarm based visual eye tracking algorithm for cheating free examination system,” *International Journal of Intelligent Systems and Applications*, vol. 5, no. 10, pp. 86–92, 2013, doi: 10.5815/ijisa.2013.10.11.

[10] M. M. Masud, K. Hayawi, S. S. Mathew, T. Michael, and M. El Barachi, “Smart online exam proctoring assist for cheating detection,” in *Advanced Data Mining and Applications*, 2022, pp. 118–132, doi: 10.1007/978-3-030-95405-5_9.

[11] R. Bawarith, A. Basuhail, A. Fattouh, and S. Gamalel-Din, “E-exam cheating detection system,” *International Journal of Advanced Computer Science and Applications*, vol. 8, no. 4, 2017, doi: 10.14569/ijacsa.2017.080425.

[12] K. Jalali and F. Noorbehbahani, “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)*, Tehran, Iran, 2017, pp. 1–6.

[13] M. Geetha, R. S. Latha, S. K. Nivetha, S. Hariprasath, S. Gowtham, and C. S. Deepak, “Design of face detection and recognition system to monitor students during online examinations using machine learning algorithms,” in *2021 International Conference on Computer Communication and Informatics (ICCCI)*, IEEE, 2021, pp. 1–4, doi: 10.1109/ICCCI50826.2021.9402553.

[14] J. O.-Rhein, B. Küppers, and U. Schroeder, “An application to discover cheating in digital exams,” in *Proceedings of the 18th Koli Calling International Conference on Computing Education Research*, 2018, pp. 1–5, doi: 10.1145/3279720.3279740.

[15] F. Ozdamli, A. Aljarrah, D. Karagozlu, and M. Ababneh, “Facial recognition system to detect student emotions and cheating in distance learning,” *Sustainability*, vol. 14, no. 20, 2022, doi: 10.3390/su142013230.

[16] I. N. Yulita, F. A. Hariz, I. Suryana, and A. S. Prabuwono, “Educational innovation faced with covid-19: deep learning for online exam cheating detection,” *Education Sciences*, vol. 13, no. 2, 2023, doi: 10.3390/educsci13020194.

[17] O. El Hammoumi, F. Benmarrakchi, N. Ouherrou, J. El Kafi, and A. El Hore, “Emotion recognition in e-learning systems,” in *2018 6th International Conference on Multimedia Computing and Systems (ICMCS)*, 2018, pp. 1–6, doi: 10.1109/ICMCS.2018.8525872.

[18] M. A. Ozdemir, B. Elagoz, A. Alaybeyoglu, R. Sadighzadeh, and A. Akan, “Real time emotion recognition from facial expressions using cnn architecture,” in *2019 Medical Technologies Congress (TIPTEKNO)*, IEEE, 2019, pp. 1–4, doi: 10.1109/TIPTEKNO.2019.8895215.

[19] E. Pranav, S. Kamal, C. S. Chandran, and M. H. Supriya, “Facial emotion recognition using deep convolutional neural network,” in *2020 6th International Conference on Advanced Computing and Communication Systems, ICACCS 2020*, 2020, pp. 317–320, doi: 10.1109/ICACCS48705.2020.9074302.

[20] A. Zbaida, M. Kodad, Y. Mohamed, and N. Benmoussa, "Deep learning and emotion recognition for an e-learning platform," *Research Gate*, pp. 1–11, 2023.
[21] W. Lu, J. Li, J. Wang, and L. Qin, "A CNN-biLSTM-AM method for stock price prediction," *Neural Computing and Applications*, vol. 33, no. 10, pp. 4741–4753, 2021, doi: 10.1007/s00521-020-05532-z.
[22] M. Coskun, A. Ucar, O. Yildirim, and Y. Demir, "Face recognition based on convolutional neural network," in *2017 International Conference on Modern Electrical and Energy Systems (MEES)*, Nov. 2017, pp. 376–379, doi: 10.1109/MEES.2017.8248937.
[23] V. Atliha and D. Sesok, "Comparison of VGG and resnet used as encoders for image captioning," in *2020 IEEE Open Conference of Electrical, Electronic and Information Sciences (eStream)*, IEEE, 2020, pp. 1–4, doi: 10.1109/eStream50540.2020.9108880.
[24] A. K. Dubey and V. Jain, "Automatic facial recognition using VGG16 based transfer learning model," *Journal of Information and Optimization Sciences*, vol. 41, no. 7, pp. 1589–1596, 2020, doi: 10.1080/02522667.2020.1809126.
[25] N. Hasan, Y. Bao, A. Shawon, and Y. Huang, "DenseNet convolutional neural networks application for predicting covid-19 using ct image," *SN Computer Science*, vol. 2, no. 5, 2021, doi: 10.1007/s42979-021-00782-7.
[26] H. Abdo, K. M. Amin, and A. M. Hamad, "Fall detection based on retinanet and mobilenet convolutional neural networks," in *2020 15th International Conference on Computer Engineering and Systems (ICCES)*, 2020, pp. 1–7, doi: 10.1109/ICCES51560.2020.9334570.
[27] Ü. Atila, M. Uçar, K. Akyol, and E. Uçar, "Plant leaf disease classification using efficientnet deep learning model," *Ecological Informatics*, vol. 61, 2021, doi: 10.1016/j.ecoinf.2020.101182.

BIOGRAPHIES OF AUTHORS

Sara Ouahabi is a Habilitated Professor (PH) at Department of Mathematics and Computer Science of Hassan II University and member of Computer Science and Information Processing Laboratory at Faculty of Science Ben M'sik. Her research includes artificial intelligence, computer security, semantic web, E-learning, computer communications (networks), and the internet of things (IoT). She conducts cutting-edge research in these areas while actively engaging in higher education to educate the next generation of computer professionals. She can be contacted at email: s3.ouahabi@gmail.com.

Rihab Aboudihaj received the B.Sc. degree in mathematical sciences and computer science from the Faculty of Sciences Ben M'Sick, Hassan II University of Casablanca, Morocco, in 2022, and the M.Sc. degree in data science and big data from Hassan II University of Casablanca, in 2024. Her research interests include the exploration and analysis of data, deep learning and machine learning algorithms, and computer vision applications. She focuses particularly on developing and improving classification and prediction models, as well as visualizing complex data to uncover useful insights and patterns. She can be contacted at email: aboudihajrihab@gmail.com.

Nawal Sael received the engineering degree in software engineering from ENSIAS, Morocco, in 2002. She has been a teacher-researcher, since 2012, an Authorized Professor, since 2014, and a professor of higher education with the Department of Mathematics and Computer Science, Faculty of Sciences Ben M'Sick, Hassan II University of Casablanca, Casablanca, since 2020. Her research interests include data mining, educational data mining, machine learning, deep learning, and the internet of things. She can be contacted at email: nawal.sael@univh2c.ma.

Kamal El Guemmat is Professor at ENSET, Hassan II University of Casablanca. He is the coordinator of the engineering cycle (computer engineering, cyber security and digital trust II-CCN). He has several skills in semantic indexing, search engines, natural language processing, digital learning, web mining, text mining, information extraction, data science, artificial intelligence, networking, data mining, ontologies, machine learning, and big data. He can be contacted at email: k.elguemmat@gmail.com.