JST: Smart Systems and Devices
Volume 34, Issue 3, September 2024, 001-008

Enhance Massive Open Online Courses Integrity: AI for Exam Proctoring

Tuan Linh Dang**, Dinh Minh Vu, Ngoc Dung Pham, The Vu Nguyen, Dinh Phu Mac, Nguyen Minh Nhat Hoang, Huy Hoang Pham
Hanoi University of Science and Technology, Ha Noi, Vietnam
**Corresponding author email: linhdt@soict.hust.edu.vn*

Abstract

Massive Open Online Courses (MOOCs) are growing quickly, but it's challenging to ensure academic integrity during remote exams with many participants. Existing approaches to supervising students online have scalability, accuracy, and integration limitations. This paper proposes a scalable, accurate AI exam proctoring module compatible with MOOCs to address this issue. Our approach includes an AI server that handles image processing and coordinates cheating detection services. Another server uses Triton to analyze student image feeds quickly. It runs optimized deep learning models, such as face recognition. There is also an integrated MOOC client to capture, compress, and transmit images. The main innovations are the asynchronous AI server for handling multiple tasks simultaneously, efficient deep learning pipelines that use fewer computing resources, and the integration of inference pipelines into Triton for faster processing. The integration of the AI module into the MOOC has been successful. The system can monitor multiple test-takers at the same time and accurately detect any potential cheating. Evaluations showed high accuracy of different AI models.

Keywords: AI, face recognition, phone detection, face pose estimation, online proctoring, MOOCs platform.

1. Introduction

The COVID-19 pandemic caused a worldwide shift to online education, affecting more than 1.6 billion learners due to campus closures [1]. Educational institutions have swiftly transitioned to remote learning methods. Massive Open Online Courses (MOOCs) emerged as a crucial alternative, experiencing massive growth as learners sought affordable and flexible options. Top MOOC providers enrollment increased from 110 million in 2019 to 180 million in 2020 [2]. Individual course enrollments frequently exceeded 100,000, making traditional supervised exams logistically impossible at this scale [3].

Automated online proctoring has become essential for credentialing and accreditation in MOOCs. However, current approaches struggle to balance scalability, reliability, privacy, and accessibility for massive, diverse learner cohorts. Existing systems rely heavily on live proctors monitoring feeds in real-time, which only scale cost-effectively beyond a few hundred students. As class sizes grow, hiring costs rise, and quality declines. There are also reasonable privacy concerns around continuous in-home recording.

The online exam monitoring system is a system that supports lecturers in taking attendance and detecting fraud actions. Violations during the exam include cases of impersonating students, they have unusual head directions or using phones. We have created an automatic monitoring system including the above features and have integrated them into the online exam system. We used hardware such as webcams on students 's laptops to capture images and send them to the AI module. From the images and information provided by students, if the system detects a violation, the evidence will be saved and sent to the lecturers.

Automated AI systems may be able to help more test takers, but currently, fully automated proctoring is not dependable. Students are frequently unjustly accused because of false positives. Developing optimized online proctoring tailored to remote contexts is still an open problem.

There is a need for optimized architectures that can support integrity for massive concurrent exams. A secure, privacy-focused automated system must monitor, flag anomalies, and escalate issues across thousands of feeds in real-time. The aim is to prevent misconduct in large-scale assessments while respecting the rights and minimizing unnecessary monitoring of students worldwide. In our knowledge, *daotao.ai* [4] is one of the first MOOC systems in Vietnam, with a large user base. However, it still does not have automated proctoring capabilities. Therefore, in this work, we introduce our proctoring structure, which is:

- Compatible with MOOCs platforms such as *daotao.ai;*
- Capable of handling many users simultaneously;

ISSN 2734-9373
https://doi.org/10.51316/jst.176.ssad.2024.34.3.1
Received: Dec 5, 2023; revised: Apr 24, 2024
accepted: Jun 25, 2024

- Highly accurate processing.

This paper is presented as follows: Section 2 shows the related work. Section 3 introduces the planned MOOCs that include an AI module. Section 4 presents the experimental results. Our manuscript concludes with Section 5.

2. Related Works

2.1 MOOCs Platforms

MOOCs have become a popular solution, providing a variety of online educational amenities, quality courses, and classroom management tools. MOOCs are designed to meet the needs of multiple students simultaneously, making it easy and convenient to transfer knowledge remotely, allowing students to access learning materials remotely and at their own pace. EdX and Coursera are two of the popular learning platforms built on the idea of MOOCs, leveraging the MOOCs model to provide a wealth of educational opportunities to learners worldwide. In Vietnam, School of Information and Communications Technology at Hanoi University of Science and Technology has developed a MOOCs platform called *daotao.ai*, which is built on the architecture of Open edX. As one of the first online learning platforms in the country, *daotao.ai* signifies a significant step forward in making education more accessible and inclusive.

At the moment, AI applications are developed to integrate into the operations of online learning platforms. The Coursera platform recently announced AI-powered tools and features that are currently in development and targeted to launch [5]. Previous studies on the application of AI in MOOCs [6-8] have extensively focused on monitoring the learning process, leaving a gap in the research regarding the use of technology during examinations. Our work aims to fill this void by concentrating on aspects such as student authentication and cheating detection during the exam phase. Despite the numerous advantages offered by MOOCs platforms, one critical area that requires attention is the lack of functionality for automatic exam monitoring. Ensuring the integrity and fairness of assessments in an online environment is paramount, particularly in light of challenges such as cheating, impersonation, and unauthorized use of external resources during exams. Addressing these concerns necessitates a concerted effort to explore AI algorithms capable of detecting cheating behaviors in exams effectively. These algorithms can analyze various parameters, including having someone else take the exam, using phones, or looking away from the camera.

By integrating AI-driven solutions for exam monitoring, MOOCs platforms can significantly enhance the reliability, objectivity, and accuracy of the evaluation process. Moreover, implementing such technologies not only serves to minimize the negative consequences associated with cheating but also reinforces the credibility and trustworthiness of online education systems.

In conclusion, while MOOCs platforms have revolutionized the landscape of education by offering unprecedented access to learning resources, addressing challenges related to exam integrity remains imperative. Through the integration of AI algorithms for detecting cheating behaviors, these platforms can uphold the principles of academic honesty and ensure a fair and equitable assessment environment for all learners.

2.2. Background

2.2.1. Face detection

Face detection serves as a foundational component in any face recognition system, playing a crucial role in accurately identifying and localizing faces within images or video frames. Over the years, deep neural networks have emerged as powerful tools for face detection, demonstrating remarkable capabilities in recognizing human faces across various contexts and conditions. However, there are some trade-offs between accuracy and speed. For instance, the Multi-task Cascaded Convolutional Networks (MTCNN) excel in speed but may sacrifice accuracy to some extent. On the other hand, models like YOLOv5-face offer flexibility in choosing between speed-focused or accuracy-focused backbones but may lack optimal balance between the two. Since face recognition needs high accuracy in face detection, we adopted RetinaFace [9], which has a high accuracy score but runs much slower than MTCNN and YOLOv5-face. Such problems can be overcome by using a lighter backbone, typically MobileNet [10].

MobileNet is a class of efficient models initially designed for mobile and embedded vision applications. The intuition behind the successful architecture of MobileNet is depth-wise separable convolution, which is proclaimed to have less computation than regular convolutions in exchange for only a negligible drop in accuracy. By integrating MobileNet as the backbone network for RetinaFace, the face detection system gains the dual advantages of computational efficiency and high accuracy. This strategic integration empowers the model to operate in real-time scenarios, making it particularly suitable for applications like MOOCs where timely processing of facial data is essential.

2.2.2. Facial feature extraction

One such approach employed in this research involves the utilization of an Improved ResNet architecture trained with ArcFace Loss [11] to generate facial embeddings with enhanced discriminative power and robustness.

ArcFace loss [11] enhances face recognition by incorporating an angular margin term into the loss function. By effectively separating feature representations in the embedding space, it brings similar features closer and separates different features, which helps the model recognize similar faces better.

Improved ResNet [12] has three main improvements over the original ResNet to train deeper networks without any accuracy drops. The first key enhancement introduced in Improved ResNet is the adoption of a diversified network architecture comprising three distinct types of Residual Blocks (ResBlocks). The design of each ResBlock varies depending on its position on the stage. This architectural refinement not only enhances the model's capacity to capture intricate facial features but also contributes to better gradient flow and more stable training dynamics. Next, the authors of Improved ResNet propose an enhanced projection shortcut mechanism that minimizes information loss during feature transformation. This improved shortcut connection facilitates smoother gradient backpropagation, enabling the network to effectively leverage information from earlier layers while mitigating the risk of vanishing gradients or feature degradation. Finally, the authors built blocks that significantly improve spatial channels for learning spatial solid patterns.

2.2.3. Face pose estimation

We must use a face pose estimation model to create the face pose detection module. This model will estimate three types of face angles: yaw, pitch, and roll. Hopenet [13] was chosen as a standout choice due to its landmark-free design, which circumvents the need for costly landmark detection models while maintaining high accuracy. This approach also reduces the server's computational load, eliminating the need for landmark-based calculations. By leveraging a convolutional neural network (CNN) architecture, Hopenet analyzes cropped face images and predicts three crucial face angles: yaw, pitch, and roll. The model achieves this feat by employing a combination of classification and regression loss functions tailored to each angle. The utilization of cross-entropy loss for angle range prediction and Mean Squared Error (MSE) loss for angle measurement ensures robust and precise estimation of face poses, thereby enhancing the overall efficacy of facial analysis tasks.

2.2.4. Object detection

In parallel, the object detection module, particularly in the context of phone detection, we use YOLO [14] (You Only Look Once) in the Phone Detection Module. YOLO is a state-of-the-art solution renowned for its accuracy, efficiency, and real-time performance. YOLOv7 is an improved version of YOLO that has better object detection. YOLOv7 introduces several key enhancements aimed at enhancing shape and size recognition, including the utilization of nine anchor boxes to improve localization accuracy. Localization accuracy is enhanced by focal loss, resulting in improved detection of small objects and overall performance. YOLOv7's higher resolution contributes to improved accuracy, and its impressive processing speed makes it a preferred choice for this application as the preferred choice for phone detection within MOOCs systems. These models can handle real-time fps computation but have not been utilized in a MOOCs system.

3. Proposed Approach

We created an AI module to be added to a MOOCs platform. It uses computer vision to help with online proctoring and learning supervision.

3.1. General Design

The proposed online proctoring module has a distributed, modular architecture that enables flexibility, scalability and fault tolerance. Fig. 1 illustrates three crucial parts: the student and teacher interfaces, the AI Server, and the Triton Inference Server [15]. The modular design allows these elements to deliver scalable and reliable online exam proctoring capabilities.

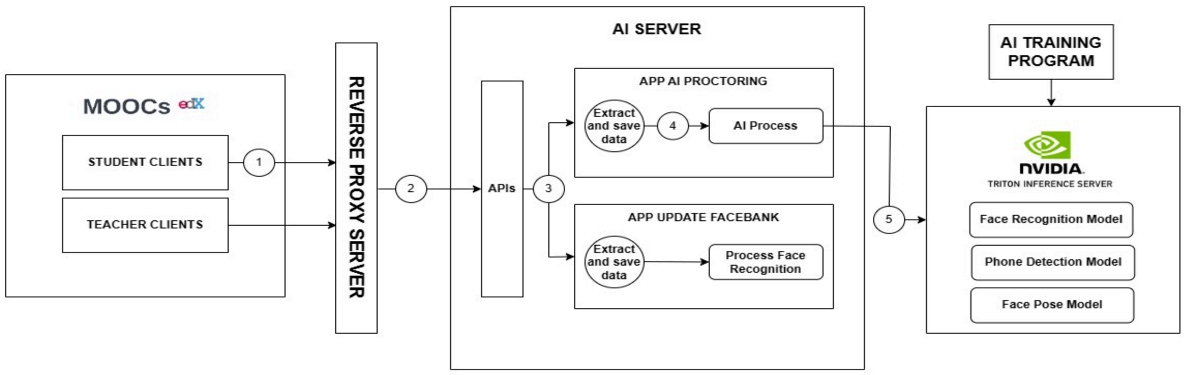

Fig. 1. Architecture of the proposed online proctoring module

The AI Server is the central part of the system. It processes images, provides AI services, and updates the database. It works as a central manager, receiving client requests and directing them to appropriate components. The Nginx reverse proxy server effectively distributes requests among AI Server workers to efficiently handle high volumes. The AI Server runs separate applications for each feature to ensure integrity and fault tolerance.

The second component is the NVIDIA Triton Inference Server, which optimizes AI capabilities, especially for deep learning models. Triton effectively manages models while maximizing hardware utilization. Three models are used for proctoring, including facial recognition for attendance, phone detection, and face pose estimation to detect suspicious behavior.

The third component comprises the student and teacher clients embedded in the MOOCs platform. For easy integration, the clients are Xblock packages installed as course modules. The student client asks to access the test-taker's camera, stream video, and request analytics from the AI Server before displaying results. The teacher-client shows proctoring outcomes for monitoring.

This distributed, modular architecture maximizes throughput and speed and handles high concurrent volumes. Asynchronous processing improves concurrency between components. The load balancer evenly distributes requests among server applications. Triton's "model instances" allow high concurrency for inferences. It works as a central manager, receiving client requests and directing them to appropriate components. The Nginx reverse proxy server distributes requests evenly across AI Server workers to handle high volumes. The AI Server runs separate applications for each feature to ensure integrity and fault tolerance.

3.2. AI Module

3.2.1. Dataset

In the face recognition module, we used a custom dataset based on the Asian-Celeb, LFW, and MS-Celeb-1M datasets for training.

We also collect additional data to test our recognition and cheating detection modules. Users will be required to perform cheating and non-cheating actions. The system will automatically capture the videos and send them back to the system. Data is labeled based on the requirements of the usage operation.

In the phone detection module, we collected 904 photos of cell phones. Our research group found the pictures online or captured them from different angles. Additionally, we extracted images from the COCO dataset, focusing on phone objects such as bags, suitcases, frisbees, snowboards, kites, surfboards, sandwiches, chairs, laptops, remote controls, keyboards, and books. We simplified the labels into two categories: "phone" and "other objects." These labels help to identify phones and avoid confusion with other objects accurately.

In the face pose detection module, we utilized the training set from the collected dataset, consisting of 1839 videos, of which 1092 videos have normal face pose directions and 742 videos have suspicious face pose directions.

3.2.2. Face recognition

The overview of the face recognition module pipeline is illustrated in Fig. 2. In the face recognition module (A), the model employs RetinaFace with MobileNets backbone to achieve high facial recognition accuracy and quickly determine facial landmarks' positions. In module B, the face alignment is done using five specific facial landmark points from the previous stage to align the face using an affine transformation.

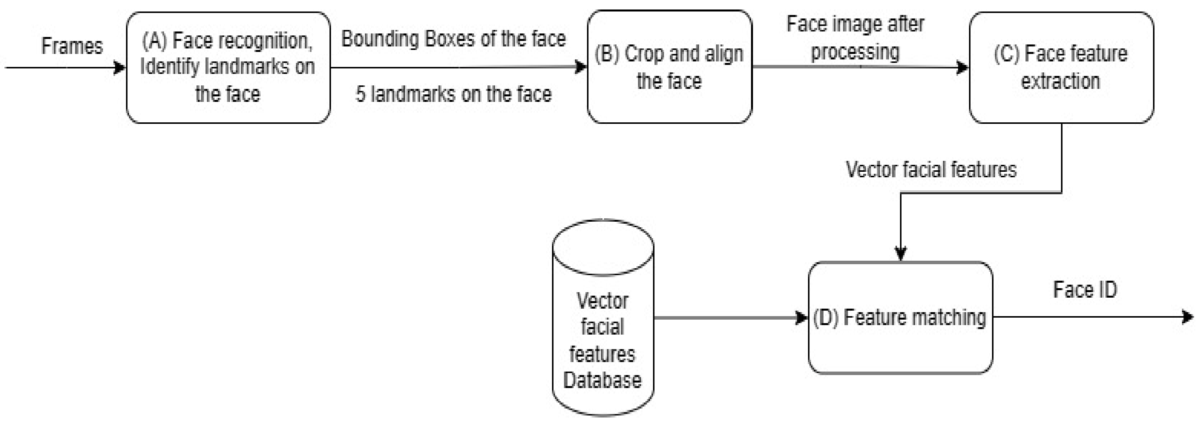

Fig. 2. Face recognition module

Within the face feature extraction module (C), the model from face.evoLVe is well-suited due to its fine-tuning on Asian face datasets. This model utilizes the IR-50 network [12], originally trained on the MS-Celeb-1M dataset using the Focal loss function. Additionally, it undergoes further fine-tuning on the face.evoLVe face dataset employing the ArcFace loss function. Finally, in the feature matching module (D), FAISS (Facebook AI Similarity Search) has been adopted for the K-NN algorithm to optimize speed when working on a large feature set. These components were trained to recognize Asian faces using a custom dataset based on the Asian-Celeb, LFW, and MS-Celeb-1M datasets. The cleaning process involved the application of the SDD-FIQA method.

3.2.3. Phone detection

Model YOLOv7 has been adopted in the phone detection module. YOLOv7 excels at precise detection, especially with small objects. It handles real-time processing at high speeds and can process high-resolution images effectively. However, the original version of YOLOv7 could perform better in phone detection. Therefore, we will retrain it with specific data to optimize performance.

3.2.4 Face pose estimation

The overview of the face pose detection module used in the application is shown in Fig. 3.

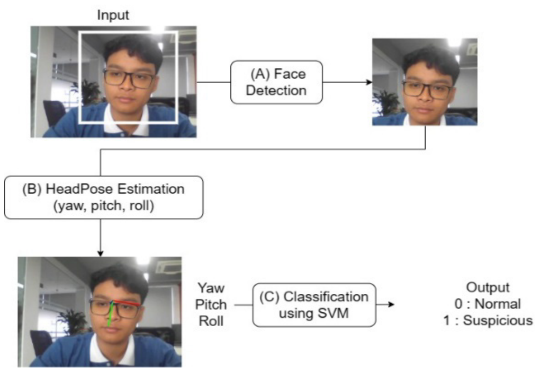

Fig. 3. Face pose estimation

In part (A), the RetinaFace model is reused for face detection. In part (B), the model Hopenet has been utilized to determine the three face pose angles (Yaw, Pitch, and Roll). Finally, in part (C), the three face pose angle values (Yaw, Pitch, and Roll) are fed into an SVM classifier to classify whether the face direction is normal or suspicious. The SVM algorithm handles noise well, ensuring high classification accuracy and very fast processing speed which is compatible with the data of face pose estimation.

The input image data is passed through blocks (A) and (B) to generate the face pose angles (Yaw, Pitch, and Roll) of the learner. These three values, along with the label indicating whether the face pose direction is normal or suspicious, are used to train the classifier (C).

3.3. MOOCs Integration

The integration of the proposed AI module to a MOOCs platform is achieved through the server-client communication structure. To maintain the continuity of MOOCs platform, the MOOCs clients were built to seamlessly attach to other MOOCs components such as examinations and lectures. The clients played the role of “sensor” for the AI module, which captured the information from the student session and transferred it to the AI Server for processing.

The HTML/CSS/JS source code will be integrated into XBlock, thereby being displayed as part of the Web page. XBlock is an API of open edX (an open-source project, developed by edX to build and develop an online education platform) which is developed with Python language.

The student clients and teacher clients are constructed with two main parts: the frontend client interface and the backend processing located on MOOCs server. Fig. 4 shows the "Face Registration " of the client interface. The "Attendance Check" and "Teacher Approval" of the teacher interface can be seen in Fig. 5 and Fig. 6. And during a proctoring session, the communication among the three objects, the student, teacher and AI module is operated as in Fig. 7 and as following procedure:

1) The client interface communicates to its backend component on MOOCs server for information retrieval and authentication;

2) The client interface captures images from the student camera, compresses the image object and sends it to AI Server along with the student’s information and lecture session identification;

3) After receiving enough information about the student clients and lecture session, as well as authentication verified, the AI Server starts processing the received image chunk for AI functions such as face recognition and phone detection;

4) The AI Server then responds to the processing result back to the client interface;

5) The client interface receives the information, then asks the backend component to save data to MOOCs server. Concurrently, it validates the AI processing result and displays it on the screen for students and teachers.

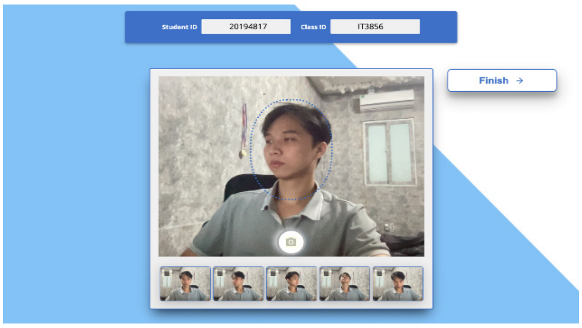

Fig. 4. Face Registration Interface

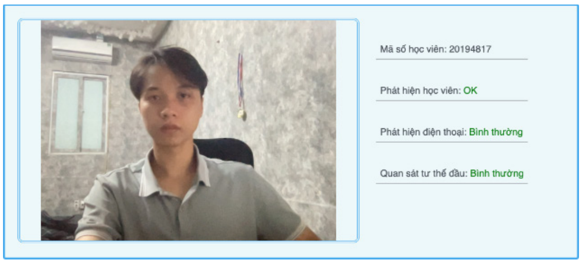

Fig. 5. Attendance Check Interface

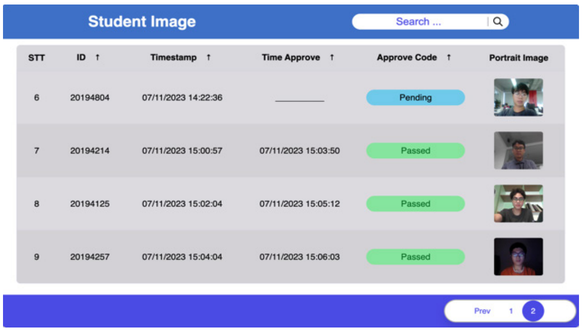

Fig. 6. Teacher Approval Interface

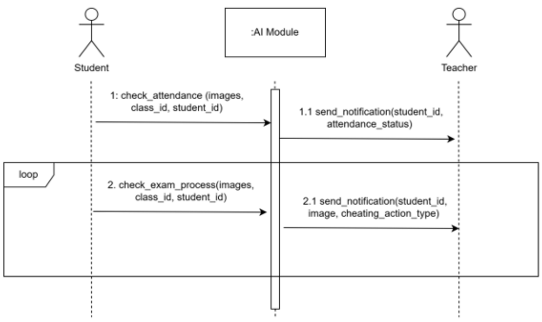

Fig. 7. System Sequence Diagram describes attendance and exam monitoring activities

Along with the communication procedure, the conversation trigger strategy also plays an important part in maintaining the performance and functionality for both AI Server and MOOC's Server. Since one of the main goals to be achieved for the module is concurrency and the ability to handle multiple sessions at the same time, the communications are designed to minimize the workload at the same time while maintaining the accuracy for AI inference. The conversation starts when a student client sends a request with images chunk and other information to the AI server for proctoring; however, the requests are not sent continuously right after the former request is sent. With the fact that a cheating action from a student may exist for longer than a few seconds, it is negligible to expand the stop time between adjacent requests to make room for other requests from other students to take place. The student clients will capture images every 0.5 seconds and the client will send all those images to the AI module every three seconds.

4. Experiment Result

4.1. Environment Setting and Parameters

4.1.1. Evaluation parameters

Since three problems are all classification problems, we will evaluate them on 4 evaluation metrics: Accuracy, Precision, Recall, F1-Score. Besides, for the face pose detection model, we use a confusion matrix to evaluate the model's performance. Additionally, for the face recognition model, we also evaluated the running time of the algorithm.

4.1.2. Environment setting

We performed training and evaluation on a personal computer with the configuration and library environment as shown in Table 1.

Table 1. Model training and testing environment

OS	Ubuntu 20.04 LTS
CPU	AMD Ryzen 3960X 24-Core Processor
GPU	NVIDIA RTX A4000
RAM	64GB DDR4

4.2. Experimental Results

4.2.1. Face recognition model

The evaluation results are based on a set of 1570 collected videos obtained during the deployment process. The results of the testing comparing our face authentication model are shown in Table 2.

Table 2. Testing result of face recognition model

Accuracy	Precision	Recall	F1-score	Run time
0.98	0.99	0.97	0.97	1164s

The results show that our model has worked well when integrated with the MOOCs with accuracy 98%. Additionally, for the face recognition model, we also evaluated the running time of the algorithm.

4.2.2. Phone detection model

The results of the evaluation depend on a dataset of 2595 videos. The results of the phone recognition model testing process are shown in Table 3.

Table 3. Testing result of phone detection model

Accuracy	Precision	Recall	F1-score
0.88	0.91	0.85	0.86

4.2.3 Face pose estimation model

After evaluating the test set of the MOOC Facepose dataset consisting of 796 short videos with 2 labels: correct face pose direction and incorrect face pose direction, we obtained the results of the confusion matrix shown in Table 4 and the results of the evaluation parameters can be seen in Table 5.

Our face pose estimation model gives quite good classification results (over 90% on all 4 metrics) on our data set and has good classification results. On both labels is the identification of "normal viewing direction" and "suspicious viewing direction"

Table 4. Result of confusion matrix of face pose estimation model

	Predicted: NO	Predicted: YES
Actual: NO	278	45
Actual: YES	23	450

Table 5. Testing result of face pose estimation model

Accuracy	Precision	Recall	F1-score
0.91	0.92	0.91	0.91

4.2.4 Practical implementation and system load

We utilized Apache Jmeter [16], a tool for load testing, to analyze and measure the performance of various services by simulating a scenario where users continuously send requests to the system. In our setup, we configured 200 users to send six photos every three seconds to the AI module, mimicking the conditions of a typical 90-minute exam. However, to rigorously test the system's capacity, we extended this simulation to 120 minutes. Upon completion of this testing phase, we confirmed that all user requests were successfully processed, and instances of detected cheating were recorded in the module's database.

Additionally, we conducted a practical demonstration of the AI Proctoring module in several real classroom settings, each with approximately 50 students. This live implementation provided tangible results on system load and functionality. Fig. 8 also illustrates how students engaged with our implemented approach during the exams

Fig. 8. Student uses our system in experiment

5. Conclusion

This paper proposes an approach to integrate an AI module into MOOC for online proctoring. There are three different models that have been investigated: face recognition, phone detection, and face pose estimation. To increase efficiency, the Triton-based server is deployed to handle multiple concurrent tasks. The results showed that we have implemented our approach successfully. The accuracies of our models in MOOCs were also good with 0.98, 0.88, and 0.91 for face recognition model, phone detection model, and face pose estimation, respectively.

A possible avenue for future research is to improve the accuracy of our models. The operation of the proposed approach is also needed to investigate with a more significant number of learners simultaneously. We also need to increase the speed of the system for concurrent users.

References

[1] Unesco, One year into covid-19 education disruption: Where do we stand?, 2021. [Online]. Available: https://www.unesco.org/en/articles/one-year-covid-19-education-disruption-where-do-we-stand Accessed on: Apr. 20, 2023.

[2] D. Shah, By the numbers: Moocs in 2020. class central, 2020. [Online]. Available: https://ww.classcentral.com/report/mooc-stats-2020/ Accessed on: Apr. 20, 2023.

[3] R. Rosmarin and J. P. 2021., The 20 most popular coursera online courses that students actually finish. [Online]. Available: https://www.businessinsider.com/guides/learning/coursera-popular-courses-withhigh-completion-rates Accessed on: Apr. 20, 2023.

[4] Daotao.ai. [Online]. Available: https://daotao.ai/ Accessed on: Apr. 20, 2023.

[5] Coursera. 2023. Coursera announces new AI content and innovations to help HR and learning leaders drive organizational agility amid relentless disruption. [Online]. Available: https://blog.coursera.org/trusted-content-and-ai-innovations-to-drive-organizational-agility-for-learning-leaders/ Accessed on: Apr. 20, 2023.

[6] A. Dubbaka and A. Gopalan, Detecting learner engagement in MOOCs using automatic facial expression recognition, 2020 IEEE Global Engineering Education Conference (EDUCON), Porto, Portugal, Apr. 27-30, 2020, pp. 447-456. https://doi.org/10.1109/EDUCON45650.2020.9125149

[7] Li, F., Zhang, X., Artificial intelligence facial recognition and voice anomaly detection in the application of English MOOC teaching system. Soft Comput, vol. 27, Apr. 2023, pp. 6855–6867. https://doi.org/10.1007/s00500-023-08119-7

[8] Sharma, K., Giannakos, M. & Dillenbourg, P., Eye-tracking and artificial intelligence to enhance motivation and learning. Smart Learn. Environ. 7, 13, 2020. https://doi.org/10.1186/s40561-020-00122-x

[9] Deng, Jiankang, *et al*. Retinaface: Single-shot multi-level face localisation in the wild. Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, Seattle, WA, USA, Jun. 13-19, 2020. https://doi.org/10.1109/CVPR42600.2020.00525

[10] Howard, Andrew G., *et al*. Mobilenets: Efficient convolutional neural networks for mobile vision applications. arXiv preprint arXiv:1704.04861, Apr. 17, 2017.

[11] Deng, Jiankang, *et al*. Arcface: Additive angular margin loss for deep face recognition, 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, Jun. 15-20, 2019. https://doi.org/10.1109/CVPR.2019.00482

[12] Duta, Ionut Cosmin, *et al*. Improved residual networks for image and video recognition. 2020 25th International Conference on Pattern Recognition (ICPR). IEEE, Milan, Italy, Jan. 10-15, 2021. https://doi.org/10.1109/ICPR48806.2021.9412193

[13] Ruiz, Nataniel, Eunji Chong, James M. Rehg. Fine-grained face pose estimation without keypoints. 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Salt Lake City, UT, USA, Jun. 18-22, 2018. https://doi.org/10.1109/CVPRW.2018.00281

[14] Wang, Chien-Yao, Alexey Bochkovskiy, Hong-Yuan Mark Liao. YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors, 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Vancouver, BC, Canada, Jun. 17-24, 2023. https://doi.org/10.1109/CVPR52729.2023.00721

[15] NVIDIA Corporation, Triton Reference Server, 2021. [Online]. Available: https://developer.nvidia.com/triton-inference-server Accessed on: Apr. 20, 2023.

[16] Apache JMeter. [Online]. Available: https://jmeter.apache.org Accessed on: Apr. 20, 2023