# A Novel Deep Learning-based Online Proctoring System using Face Recognition, Eye Blinking, and Object Detection Techniques

Istiak Ahmad[1], Fahad AlQurashi[2], Ehab Abozinadah[3], Rashid Mehmood[4]
Department of Computer Science, Faculty of Computing and Information Technology[1,2]
Department of Information Systems, Faculty of Computing and Information Technology[3]
High Performance Computing Center[4]
King Abdulaziz University, Jeddah 21589, Saudi Arabia

***Abstract*—Distance and online learning (or e-learning) has become a norm in training and education due to a variety of benefits such as efficiency, flexibility, affordability, and usability. Moreover, the COVID-19 pandemic has made online learning the only option due to its physical isolation requirements. However, monitoring of attendees and students during classes, particularly during exams, is a major challenge for online systems due to the lack of physical presence. There is a need to develop methods and technologies that provide robust instruments to detect unfair, unethical, and illegal behaviour during classes and exams. We propose in this paper a novel online proctoring system that uses deep learning to continually proctor physical places without the need for a physical proctor. The system employs biometric approaches such as face recognition using the HOG (Histogram of Oriented Gradients) face detector and the OpenCV face recognition algorithm. Also, the system incorporates eye blinking detection to detect stationary pictures. Moreover, to enforce fairness during exams, the system is able to detect gadgets including mobile phones, laptops, iPads, and books. The system is implemented as a software system and evaluated using the FDDB and LFW datasets. We achieved up to 97% and 99.3% accuracies for face detection and face recognition, respectively.**

***Keywords*—Online learning; online proctor; student authentication; face detection; face recognition; eye blinking detection; object detection; distance learning; e-learning**

## I. Introduction

Most schools and universities provide educational courses and training physically, i.e., requiring the attendance of lectures, entrance examinations, semester exams, and other activities in physical classrooms and spaces. Teaching and learning in physical spaces have many disadvantages such as inflexibility for students, teachers, and other staff, requiring physical spaces with stringent requirements, accessibility-related challenges for the students and staff in terms of space and time, challenges related to human disabilities, higher financial costs, transportation-related challenges and harms to people and environment, and many more. Online teaching and learning have been known to have many advantages including the flexibility and accessibility for people to attend classes from homes, at their convenience both in time and space, lower costs, a much smaller impact on the planet environment, and many more. Indeed, all the disadvantages of in-class teaching mentioned above could be overcome or abated by online learning.

Despite the many benefits of online learning, in-class learning has remained the mainstream choice for teaching and learning. Massive open online courses (MOOCs) that are offered online have motivated many to attend and complete courses and degrees online [1]. Many of the top schools and universities worldwide provide students with online courses as well as certificates upon completion of the courses. However, these MOOCs are mainly used to upskill knowledge rather than replace school and university education. This trend of moving towards MOOCs had been on the rise and is expected to take an increasingly larger share of in-class education.

The COVID-19 pandemic has caused disruption in many spheres of our lives. Physical interaction for education, work, and leisure has been regulated by governments around the world to minimise human infection rates [2]. This situation has forced education and many other physical activities and businesses to move from physical to online spaces [3]. School, university, and other education and training around the world have moved to online learning. However, many challenges are prohibiting its wide adoption by the governments and public. For example, monitoring of attendees and students during classes, particularly during exams, is a major challenge for online systems due to the lack of physical presence. There is a need to develop methods and technologies that provide robust instruments to detect unfair, unethical, and illegal behaviour during classes and exams. Current literature in this respect is limited with most of the software available from commercial entities that provide limited and "non-open" software tools. Many open-source tools and efforts are needed to bring innovation, variety, and richness to this online learning software systems domain.

Artificial intelligence (AI) has revolutionized our world and environments by providing smartness to many of our daily life activities [4], [5], [6], albeit with several challenges [7], [8]. Particularly, machine and deep learning has accelerated innovation in many fields such as education [1], healthcare [9], [10], transportation [11], [12], communication networks [13], disaster management [14], smart cities [15], and many more. With no exception, AI has the capacity to revolutionise online learning and proctoring.

This paper proposes a novel online proctoring system that uses deep learning to continually proctor physical places without the need for the presence of a physical proctor.

The system employs biometric approaches including face recognition using the HOG face detector and the OpenCV face recognition algorithm. Also, the system incorporates an eye blinking detection method to detect stationary pictures. Moreover, to enforce fairness during exams, the system is able to detect gadgets including mobile phones, laptops, iPads, and books. The system is implemented as a software system and evaluated using the FDDB (Face Detection Data Set and Benchmark) and LFW (Labeled Faces in the Wild) datasets.

The rest of the paper is structured as follows. Section II discusses the research related to online proctoring systems. Section III describes the methodology and design of the proposed system. Section IV provides system evaluation. Section V concludes and discusses future work.

## II. Literature Review

An overview of the relevant research is presented in this section. In the online proctoring system, Section II-A analyzes the literature in the academic realm, and Section II-B reviews the literature in the commercial sector.

### A. Academic Research

The online exam is facing immense challenges throughout the exam. Sarrayrih et al. [16] discussed the several challenges presented by the online exam, as well as providing a solution by grouping the hostnames or IPs of clients for a specific location and time, with a biometric solution like face recognition and fingerprints. In [17], a profile-based authentication framework is proposed for the online exam based on different challenging questions, including the favourite questions, personal questions, and an academic question. Fenu et al. [18] proposed a multi bio-metric continuous authentication system including face recognition, voice recognition, touch recognition, mouse, and keystroke in 2018. Selvi et al. [19] designs and implements a firewall security system using different firewall technologies, including Network Address Translation, Demilitarized Zone, and Virtual Protocol Network , which are used for intrusion detection. Wei et al. [20] proposed fingerprint-based solution. Garg et al. [21] proposed a face recognition and detection solution for the secured online exam using deep learning. Another online proctoring system was proposed by Atoum et al. [22], which continuously estimates six components, including voice, phone, text, and active window detection, gaze estimation, and user verification. A fingerprint and eye tracker-based online test management system was proposed by Bawarith et al. [23]. Cheating and not cheating are used as student status to evaluate their proposed methodology.

### B. Commercial Systems

Recently, the online proctoring system has become a challenge for researchers and developers. Due to the coronavirus situation, the demand and challenges for the online proctoring system are enormously increasing day by day. Several industrial companies developed the proctoring system commercially with the paid version. For example, Mettl [24], Proctortrack [25], Proctoredu [26], Proctoru [27], Comprobo [28], and so on.

**Mettl:** Mettl created web-based online proctoring software that divides their system into four major components: candidate authentication using a picture, OTP and ID affirmation, human-based proctoring using real-time recording in the classroom, secure browser-based proctoring using disables the following features: opening new browsers and data transferring media, and AI-based proctoring using facial, mobile phone, candidate distraction, and multiple person detection. To use this software, the examiners have to pay.

**Proctortrack:** Proctortrack built a web-based multi-level proctoring system through four levels of security. Level 1 is called ProctorLock, and it has automated identity verification, including audio, video, and desktop data recording. The real-time video data is available up to 2-3 hours for the proctor. Level 2 is named ProctorAuto, and it includes level 1 features with automated data analysis. Level 3 is called ProctorTrackQA, which is a robust version of level 2. It analyzes the results from level-2 using a manual QA review process. Finally, level 4, called ProctorLive AI, includes AI-based auto proctoring intervention capabilities in cases of suspicious reactions, cheating, or aiding a student. Testing sincerity results are additionally analyzed with AI.

**Proctoredu:** Proctoredu is a web-based online proctoring system which includes features like online supervision and video recording, additional camera, face and voice detection, passport recognition, face bio-metric, focus, and online status tracking, locking a parallel login, content copy protection, screen recording, and determining a second monitor. For PC Chrome and Firefox, all features are available. For Android, Chrome and IOS Safari supported all the features except screen recording and second monitor determination.

**Proctoru:** Proctoru promoted a web-based online proctoring system including live proctor monitoring, flagging, and intervention. Admins can observe the session in real-time and perform AI-based behavior analysis.

**Comprobo:** Comprobo solutions developed an online automated invigilation system on the web-based platform including features such as capturing the user's photographic ID against each assignment and verifying the reference photo, substitution check, restriction of using other applications or browsers, recording the IP address of the device, remote ID verification, biometric monitoring, and recording the full working environment.

Motivated by the challenges in the area of online learning systems, we propose and implement an online proctoring system that provides solutions to mitigate problems including fraud and multiple student attendance, cheating using still images, and unauthorised use of devices during class or exam time.

## III. Methodology and Design

### A. The Proposed Framework

The proposed web-based online proctoring system is distributed into two modules. Firstly, the online registration part, and secondly, the online proctoring part. Fig. 1 describes the proposed architecture for the online proctoring system.

*1) Online Registration:* For registering students' faces, we accessed the student's web-camera through HTTPS protocol

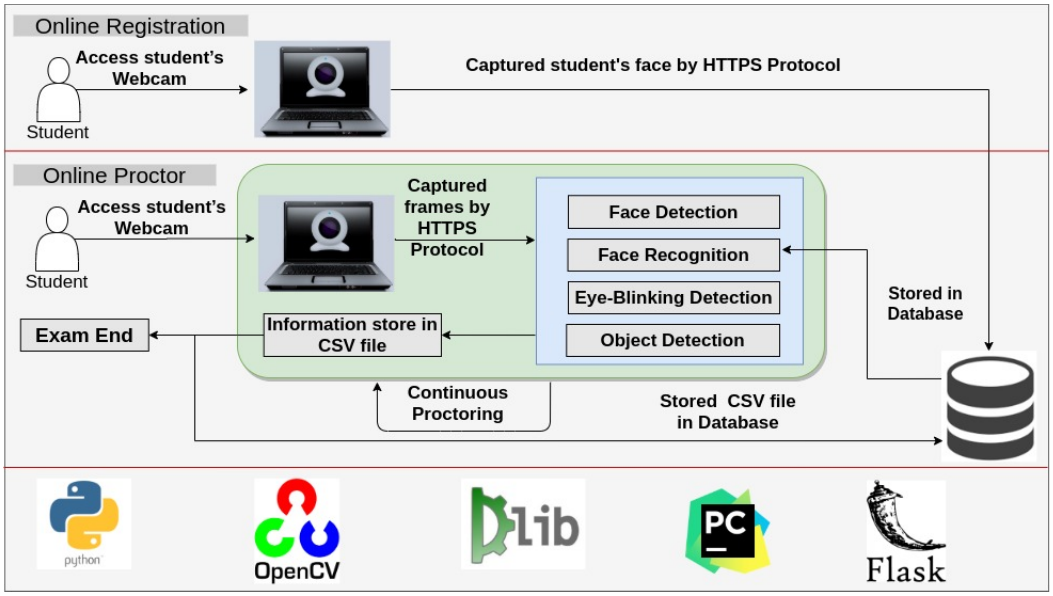


Fig. 1. Proposed Architecture.

during registration and captured the students' faces, storing the face information in the database. We used a flask micro-framework for web development.

The primary challenge in this module was accessing the client-side camera to capture the students' faces. To get around this, we utilized the HTTPS (HyperText Transfer Protocol Secure) protocol to access the students' webcams, which encrypts all conversations between the browser and the server. When using the HTTPS protocol to host a website, SSL certificates are also required. We used a self-signed SSL certificate to run the server.

*2) Online Proctoring:* During online exam sessions, there are some challenges to conducting the exam. Challenges are:

- An unauthorized student may participate in the exam.
- Multiple students may participate together for the exam.
- The student may use his still picture for face recognition.
- The student may use a device such as a mobile, laptop, or IPad to run a video for face recognition.
- The student may use books during the exam.

Our main goal is to mitigate those challenges. In the online proctoring module, we use biometric methods like face detection and recognition with eye-blinking detection. The suggested system's algorithm is detailed in Algorithm 1. In the face-recognition part, we detect and recognize students' faces and detect multiple faces in front of the camera. There is a chance that students can use their still pictures in front of the web-camera. As a result, the face-recognition algorithm recognizes the student as a real face. To avoid the recognition of still pictures, we use eye-blinking methods. If the number of eye-blinking is not more than 30, then we can confirm that the picture in front of the camera is still. There is another possibility that students can hold a device in front of the web camera by playing his face video. In this case, face recognition and eye-blinking algorithms will detect the image as real and authenticate. So, we use object detection methods like YOLOv3, which also serve to prevent cheating in the exam using devices. We also detected the book using the same YOLOv3 model as the fare and secure exam.

**Algorithm 1** Master Algorithm

```
procedure onlineProctor
    while True do
        frame ← captureFrameFromWebCamera
        faceDetect ← faceDetection(method = "HOG")
        faceCount ← detectFaceCountForCurrentFrame
        if faceCount == 0 then
            Exam cancel: no face found
        else if facecount == 1 then
            Exam continue
        else if $facecount > 1$ then
            Exam cancel: multiple face found
        end if
        faceMatch ← faceRecognition()
        if faceMatch == False then
            Exam cancel: Unauthorized face found
        end if
        blinkingRatio ← eyeBlinkingDetection()
        if $blinkingRatio > 4.5$ then
            noBlinking ← noBlinking + 1
        end if
        if $noBlinking > 30$ then
            Exam cancel: still image found
        end if
        object ← objectDetection()
        if object == list of target objects then
            Exam cancel: object found
        end if
    end while
end procedure
```

### *B. Datasets*

We have used FDDB data sets for face detection which contain 5171 face annotation from 2845 images collected from Faces in the wild data sets. We divide the data sets into two parts, including faces and without faces, and implement face detection algorithms to evaluate our proposed system. The resolution of each image in the data set is 86 x 86 pixels. To evaluate face recognition algorithms, we used LFW dataset which contain 5749 people's 13233 images, where 1680 people's had two or more images. As our face recognition algorithm needs a single image for face recognition, we divided the data sets into ten sections based on the number of images available of the people in each data set. In the data set, every 3180 people have one image, every 775 and 290 people have two and three images consecutively, and so on. Each of the image resolutions is 250 x 250 pixels.

### *C. Face Detection*

In our proposed system, we used the HOG (Histograms of Oriented Gradients) method to detect the faces that were proposed by Navneet and Dalal [29]. In the initial stage of face detection, we convert our input image into grayscale because we don't need an RGB image to find faces. After that, we process every single pixel and the directly surrounding pixels of the image at a moment. We would like to determine the darkness of the current pixel is in contrast to the pixels around it. To show in which direction the image is becoming darker, we draw an arrow. If we replicate such a method for each and every pixel in the image, then we discover that every pixel is followed by an arrow. Gradients are the arrows, which determine the overall image's movement from brightness to darkness. Following that, we can see the image's fundamental pattern. To conduct the function, we divided all of the images into 16x16 pixel squares. Then we count gradient points in each major direction of each square and replace the square image with the strongest single gradient direction. The process's output will convert the original picture into the face's fundamental structure, which seems to be the most similar to the HOG pattern derived from training images. We used the HOG frontal face detector using dlib and the OpenCV library for face detection.

### *D. Face Recognition*

Face recognition is the most popular biometric solution for the online authentication system. OpenCV is a famous computer vision library that was started by Intel in 1999. OpenCV implements three face recognition algorithms, including Eigenface, Fisherface, and LBPH (Local Binary Patterns Histograms) face recognition. To detect faces, these algorithms employ the Haar cascade classifier technique, introduced by Paul and Michael [30].

In our proposed methodology, we snap a picture of a student as input and use HOG techniques to recognize faces in the image. Then, for the identified picture, estimate the 68 landmarks. Faces that are oriented differently and seem differently to a computer may all belong to the same person, and these signs can be used to easily identify them. Finally, the identified photos are directly compared to previously learnt and saved faces in our database. The pseudo code for facial recognition is shown in the Algorithm 2.

We match a known face from our database to unknown faces using a deep neural network. We train a classifier to determine which known student is the closest match based on measures from a new test image. The classifier's output would be the name of a student. The number of faces in the photograph is also counted.

**Algorithm 2** Face Recognition

```
procedure Face Recognition (studentId, studentName)
    while True do
        Grab current frame from student's Webcam
        frame ← currentFrame
        faceLocations ← get all faces on frame
        faceEncoding ← get all faceEncodings on frame
        faceMatch ← compare studentFace with all faces
        if faceMatch == True then
            Face Recognized
        else
            Face Unrecognized
        end if
    end while
end procedure
```

*1) Facial Landmark Estimation:* To a computer, the split faces rotated in various orientations appear to be different. To address this issue, we apply the face landmark estimation algorithm [31], which aids in the localization and representation of important facial features including the right and left eye, nose, jawline, mouth, and right and left eyebrow. The HELEN dataset is being utilized to find 194 landmarks on the face from a single image in a millisecond using this approach, which gives an ensemble of randomized regression trees. The method below can help determine whether two faces facing different directions and appearing differently from a computer's perspective are actually the same person.

Based on the fundamental concept of 68 distinct places on an image, we will train the system to recognize any 68 specific landmarks from the target image. We can center the eyes and lips no matter how the faces are rotated after using this method. The landmark boundary of the face is shown in Fig. 3(a), and the face landmark with 64 points is shown in Fig. 3(b).

*2) Encode the Faces:* The most basic concept in facial recognition is matching a recognized face to an unknown one. We identify a previously tagged face that appears to be frighteningly similar to an unknown face as belonging to the same individual. If there are thousands of students, it will take a long time to recognize everyone. As a result, we will need a technique for extracting a few basic measures from each face so that we can measure our unknown face and find the closest known face. We may, for example, measure the distance between the eyes and eyebrows, the length of the nose and mouth, and the size of each ear.

### *E. Eye Blinking Detection*

To identify a still image, the eye blinking method is utilized. Each eye is represented by 6 (x, y)-coordinates, which begin in the upper left corner and work clockwise around the rest of the area.

**Algorithm 3** Eye Blinking Detection

```
procedure Blinking Detection(studentId, studentName)
    eyeModel ←load shape-predictor-68-face-landmarks
    while True do
        frame ← Grab current frame from Webcam
        # get Blinking Ratio
        function bRatio( EyePoint, landmark)
            leftPoint ← left eye point
            rightPoint ← right eye point
            centerTop ← center top eye point
            centerBottom ← center bottom eye point
            horLineLenght ← horizontal line of eye point
            verLineLenght ← vertical line of eye point
            ratio ← horLineLenght / verLineLenght
            return ratio
        end function
        faces ←dlibFrontalFaceDetector (Frame)
        for face ← faces do
            # facial landmark (lm)
            lm ← eyeModel(grayFrame, faces)
            # Left Eye Ratio
            ler ← bRatio ([37,38,39,40, 41,42], lm)
            # Right Eye Ratio
            rer ← bRatio([43,44,45,46, 47,48], lm)
            bliningRatio ← (ler + rer)/2
        end for
    end while
end procedure
```

From Fig. 2 image, we can get the key point to find the relation between height and width.

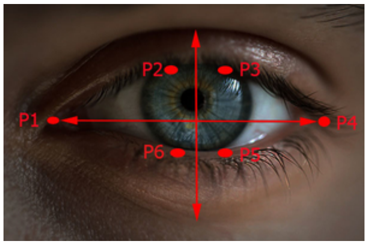


Fig. 2. Eye Blink Detection Landmarks.

In 2016, soukuter and cechj[32] proposed a method for real-time eye detection using facial landmarks. According to their research, we use an eye aspect ratio (EAR) equation that represents this connection, where 2D facial landmark positions are p1,..., p6. We utilize dlib and OpenCV to create eye blinking detection using facial landmarks and a frontal face detector.

### F. Object Detection

Object recognition refers to identifying objects from digital images. Object classification can be divided into three tasks: object localization, image classification, and object detection. Object segmentation is the final task for object recognition. R-CNN, YOLO, SSD, RetinaNet, and ImageNet are popular deep learning-based object recognition models. The improved versions of the R-CNN model are Faster R-CNN, and Fast R-CNN, which are demonstrated and designed for object recognition, and object localization. The acronym YOLO stands for 'You Only Look Once'. YOLO model versions are YOLOv2 and YOLOv3 [33].

The R-CNN family of models delivers excellent object identification accuracy, but its processing speed is a key drawback. The processing speed is just 5 frames per second on a GPU, but the YOLO model is significantly quicker than R-CNN since a single-layer neural network is applied to the entire image. The YOLOv3 model is 100 times quicker than fast R-CNN, and 1000 times quicker than R-CNN. For training, the YOLO model is linked to a single neural network. It takes pictures and divides them into a grid of cells, with the cells anticipating bounding boxes and class labels. The predicted accuracy rate for this model is lower. "YOLO9000: better, faster, stronger" is how YOLOv2 is known. This model can predict 9000 object classes after being trained on two object identification datasets in parallel. The model is trained using high-resolution input pictures and batch normalization. Darknet-19, a proprietary deep architecture with a 19-layer neural network augmented with an additional 11 layers to identify the objects, was utilized by YOLOv2. YOLOv2's 30-layer design made it difficult to detect tiny objects, but it's primarily utilized in real-time object identification when precision isn't required.

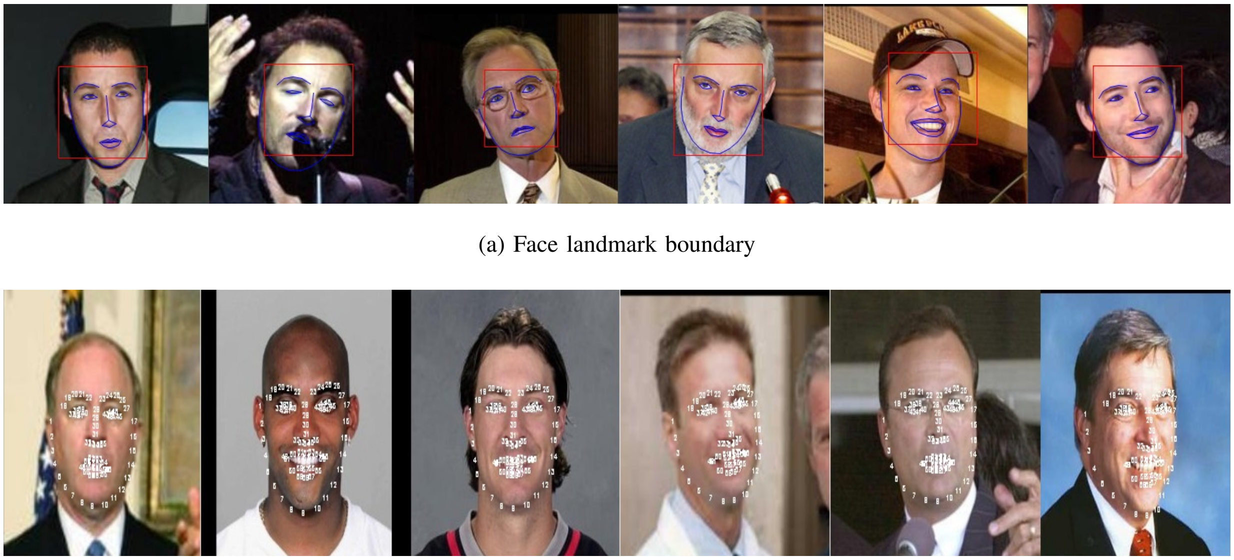
YOLOv3 is better than YOLOv2 in terms of speed and strength. It implements the darknet-53 proprietary deep architecture, which includes a 53 network and additional layers for object identification trained on ImageNet. As a result, it has a fully convolutional underlying architecture with 106 layers. The YOLOv3 model was employed in our suggested approach. This model was trained on pictures from the COCO dataset with different sizes: 608 x 608 (less speed, high accuracy), 416 x 416 (moderate speed, moderate accuracy), and 320 x 320 (high speed, low accuracy), and includes 80 labels such as laptop, mobile phone, and book.

**Algorithm 4** Object Detection Algorithm

```
procedure ObjectDetection (studentId, studentName)
    while True do
        Grab current frame from student's Webcam
        frame ←currentFrame
        height, width, channels ←frame.shape
        blob ←blobFromImage(frame, scale)
        set the blob in network as input
        outputs ← forward pass to get Outputlayer
        for out ← outputs do
            for detection ← output do
                #scan outputs to get max confidence score
                confidence ←max scores
                if $confidence > 0.7$ then
                    Object detected
                end if
            end for
        end for
    end while
end procedure
```

## IV. System Evaluation

To implement our proposed system, we use OpenCV for image processing, the dlib package for machine learning and



(a) Face landmark boundary

(b) Face landmark with 64 points

Fig. 3. Facial Landmarks.

Python, etc. We continuously proctor the exam system and concurrently implement each of the methods. Several experiments have been carried out to determine its efficiency, and the results are presented below.

We used the confusion matrix, which is a technique for summarizing the performance of a classification algorithm. There are some key terms for evaluating the confusion matrix, including TP, TN, FP, FN, accuracy, precision, recall, etc.

**True Positive (TP):** The number of positive samples that are accurately labelled.

**True Negative (TN):** The number of negative samples that are accurately labelled.

**False Positive (FP):** The number of negative samples that are mislabelled as positive.

**False Negative (FN):** The number of positive samples that are mislabelled as negative.

**Accuracy:** The percentage of classes are properly anticipated across all classes.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \times 100$$

**Precision:** The percentage of positive classes that are accurately predicted and truly positive.

$$Precision = \frac{TP}{TP + FP} \times 100$$

**Recall:** The percentage of classes are properly anticipated across all positive classes.

$$Recall = \frac{TP}{TP + FN} \times 100$$

### A. Face Detection

The face detection results are shown in Fig. 4 and 5. We have used the “HOG” method for face detection. In our test image dataset, we have a total of 4305 images with faces from the FDDB dataset. After implementing the face detection method, we got 97.21% accuracy, 100% precision, 97.18% recall, and 98.57% F1-score. Additionally, we obtained 4141, 0, 44, and 120 TP, FP, TN, and FN from the confusion matrix, correspondingly.

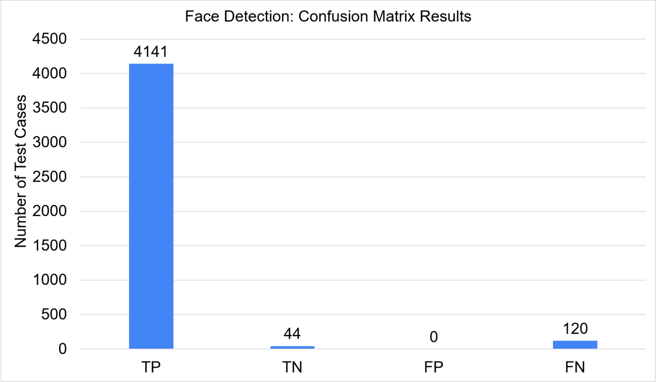


Fig. 4. Face Detection Result (TP, TN, FP, FN).

### B. Face Recognition

The experimental result for face recognition is shown in Table I, and II. We used the LFW face dataset to experiment with the results of the face recognition algorithm. In our proposed system, we need a single image per person for the proposed algorithm. So, we use one image per person in the training image, whereas in the test image set, we use the corresponding person images in identical or different backgrounds or poses, as well as unknown person mages. Table I shows the confusion matrix number of correctly and

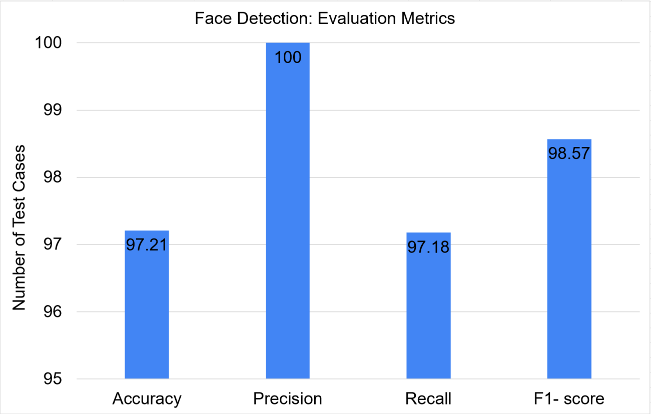


Fig. 5. Face Detection Results (Accuracy, Precision, Recall, F1-Score).

incorrectly predicted images for several users and test images using TP, TN, FP, and FN. We achieved reasonable results with the proposed algorithm.

TABLE I. Face Recognition Result (TP, TN, FP, and FN)

| No. of Users | No. of Test Cases | TP | TN | FP | FN |
|---|---|---|---|---|---|
| 3810 | 5256 | 3808 | 0 | 1410 | 38 |
| 143 | 3897 | 3824 | 0 | 52 | 21 |
| 775 | 1431 | 1401 | 0 | 28 | 2 |
| 290 | 808 | 779 | 0 | 24 | 5 |
| 187 | 695 | 675 | 0 | 11 | 9 |
| 112 | 525 | 508 | 0 | 12 | 5 |
| 55 | 304 | 298 | 0 | 3 | 3 |
| 39 | 253 | 249 | 0 | 2 | 2 |
| 33 | 247 | 245 | 0 | 2 | 0 |
| 26 | 221 | 212 | 0 | 7 | 2 |
| 15 | 144 | 143 | 0 | 1 | 0 |

We perceive from Table II that the accuracy of the algorithm is almost near to 99% for around 1000 people, but if the number of people is increased, then the accuracy decreases slowly. For example, if we train 15 to 775 people's faces, then the accuracy is near to 99% and for 143 people's faces, we used a large image set for testing where we achieved a better accuracy of about 98%. For training 3810 people's faces, we got a low accuracy result of about 72%. Based on the aforementioned findings, we may conclude that the suggested algorithm performs significantly better with a smaller number of students.

TABLE II. Face Recognition Results

| No. of Users | No. of Test Cases | Accuracy | Precision | Recall | F1 Score | Time (S) |
|---|---|---|---|---|---|---|
| 3810 | 5256 | 72.45 | 72.98 | 99.01 | 84.02 | 2373.03 |
| 143 | 3897 | 98.12 | 98.66 | 99.45 | 99.05 | 1096.74 |
| 775 | 1431 | 97.90 | 98.04 | 99.86 | 98.94 | 587.42 |
| 290 | 808 | 96.41 | 97.01 | 99.36 | 98.17 | 304.15 |
| 187 | 695 | 97.12 | 98.39 | 98.68 | 98.54 | 268.46 |
| 112 | 525 | 96.76 | 97.69 | 99.02 | 98.35 | 173.27 |
| 55 | 304 | 98.03 | 99.00 | 99.00 | 99.00 | 97.44 |
| 39 | 253 | 98.42 | 99.20 | 99.20 | 99.20 | 76.49 |
| 33 | 247 | 99.19 | 99.19 | 100.0 | 99.59 | 83.38 |
| 26 | 221 | 95.93 | 96.80 | 99.07 | 97.92 | 67.48 |
| 15 | 144 | 99.30 | 99.30 | 100.0 | 99.65 | 41.06 |

### C. Object Detection

As we mentioned earlier, we implemented the Yolov3 object detection model for our proposed system that was proposed by Joseph and Ali [33], and the model was assessed on the COCO dataset. They compared YOLOv3 with RetinaNet and found that YOLOv3 has a similar mean average precision (mAP) with a considerably quicker inference time. For example, YOLOv3–608 achieved 57.9% mAP in 51 milliseconds, while RetinaNet-101–800 achieved 57.5% mAP in 198 milliseconds, a 3.8 milliseconds faster.

## V. Conclusion and Future Work

Face recognition and object identification techniques are utilized in this study to give comprehensive knowledge for online tests. Our proposed method will aid in reducing inequity during the online exam. Human-induced detection is very important when conducting an online proctoring system, as it will aid in detecting students' suspicious behavior throughout the test. We do not incorporate human activity detection in our suggested model, instead of relying on a single biometric solution and object recognition approaches for the online proctoring system.

In the future, we hope to apply and investigate various human behaviors such as gazing out the window, conversing with people, focusing on other directions, moving about, and so on. We only utilize the YOLOv3 model because of its quicker object detection algorithms, although there are several other object detection approaches available. In the next study, we will focus on such approaches and compare them to our current suggested system.

We have evaluated our proposed system using two datasets. However, the system has not been tested in a real-life deployment with a large number of users. Future work will look into further testing and development of the system in real-life environments. The current proctoring systems are commercial and their designs and sources are not available openly. This work is an effort to develop open systems so the community can learn from each other leading to faster innovations in the field under open-source developments.

## Acknowledgment

The work reported in this paper is supported by the High Performance Computing Centre at King Abdulaziz University, Saudi Arabia. The experiments reported in this paper were performed on the Aziz supercomputer at King Abdulaziz University.

## References

[1] R. Mehmood, F. Alam, N. N. Albogami, I. Katib, A. Albeshri, and S. M. Altowaijri, "UTiLearn: A Personalised Ubiquitous Teaching and Learning System for Smart Societies," *IEEE Access*, vol. 5, pp. 2615–2635, 2017.

[2] E. Alomari, I. Katib, A. Albeshri, and R. Mehmood, "COVID-19: Detecting Government Pandemic Measures and Public Concerns from Twitter Arabic Data Using Distributed Machine Learning," *International Journal of Environmental Research and Public Health*, vol. 18, no. 1, p. 282, jan 2021. [Online]. Available: https://www.mdpi.com/1660-4601/18/1/282

[3] F. Alam, A. Almaghthawi, I. Katib, A. Albeshri, and R. Mehmood, “iResponse: An AI and IoT-Enabled Framework for Autonomous COVID-19 Pandemic Management,” *Sustainability*, vol. 13, no. 7, p. 3797, mar 2021. [Online]. Available: https://www.mdpi.com/2071-1050/13/7/3797

[4] T. Yigitcanlar, N. Kankanamge, M. Regona, A. Maldonado, B. Rowan, A. Ryu, K. C. Desouza, J. M. Corchado, R. Mehmood, and R. Y. M. Li, “Artificial Intelligence Technologies and Related Urban Planning and Development Concepts: How Are They Perceived and Utilized in Australia?” *Journal of Open Innovation: Technology, Market, and Complexity*, vol. 6, no. 4, p. 187, dec 2020. [Online]. Available: https://www.mdpi.com/2199-8531/6/4/187

[5] T. Yigitcanlar, L. Butler, E. Windle, K. C. Desouza, R. Mehmood, and J. M. Corchado, “Can Building “Artificially Intelligent Cities” Safeguard Humanity from Natural Disasters, Pandemics, and Other Catastrophes? An Urban Scholar’s Perspective,” *Sensors*, vol. 20, no. 10, p. 2988, may 2020. [Online]. Available: https://www.mdpi.com/1424-8220/20/10/2988

[6] N. Janbi, I. Katib, A. Albeshri, and R. Mehmood, “Distributed Artificial Intelligence-as-a-Service (DAIaaS) for Smarter IoE and 6G Environments,” *Sensors*, vol. 20, no. 20, p. 5796, oct 2020. [Online]. Available: https://www.mdpi.com/1424-8220/20/20/5796

[7] T. Yigitcanlar, J. M. Corchado, R. Mehmood, R. Y. M. Li, K. Mossberger, and K. Desouza, “Responsible Urban Innovation with Local Government Artificial Intelligence (AI): A Conceptual Framework and Research Agenda,” *Journal of Open Innovation: Technology, Market, and Complexity*, vol. 7, no. 1, p. 71, feb 2021. [Online]. Available: https://www.mdpi.com/2199-8531/7/1/71

[8] T. Yigitcanlar, R. Mehmood, and J. M. Corchado, “Green Artificial Intelligence: Towards an Efficient, Sustainable and Equitable Technology for Smart Cities and Futures,” *Sustainability 2021, Vol. 13, Page 8952*, vol. 13, no. 16, p. 8952, aug 2021. [Online]. Available: https://www.mdpi.com/2071-1050/13/16/8952

[9] T. Muhammed, R. Mehmood, A. Albeshri, and I. Katib, “UbeHealth: A personalized ubiquitous cloud and edge-enabled networked healthcare system for smart cities,” *IEEE Access*, vol. 6, pp. 32 258–32 285, 2018.

[10] S. Alotaibi, R. Mehmood, I. Katib, O. Rana, and A. Albeshri, “Sehaa: A Big Data Analytics Tool for Healthcare Symptoms and Diseases Detection Using Twitter, Apache Spark, and Machine Learning,” *Applied Sciences*, vol. 10, no. 4, p. 1398, feb 2020. [Online]. Available: https://www.mdpi.com/2076-3417/10/4/1398

[11] E. Alomari, R. Mehmood, and I. Katib, “Road Traffic Event Detection Using Twitter Data, Machine Learning, and Apache Spark,” in *2019 IEEE SmartWorld, Ubiquitous Intelligence & Computing, Advanced & Trusted Computing, Scalable Computing & Communications, Cloud & Big Data Computing, Internet of People and Smart City Innovation (SmartWorld/SCALCOM/UIC/ATC/CBDCom/IOP/SCI)*. Leicester, UK: IEEE, aug 2019, pp. 1888–1895. [Online]. Available: https://ieeexplore.ieee.org/document/9060213/

[12] M. Aqib, R. Mehmood, A. Alzahrani, I. Katib, A. Albeshri, and S. Altowaijri, “Smarter traffic prediction using big data, in-memory computing, deep learning and gpus,” *Sensors (Switzerland)*, vol. 19, no. 9, 2019.

[13] T. Mohammed, A. Albeshri, I. Katib, and R. Mehmood, “UbiPriSEQ—Deep reinforcement learning to manage privacy, security, energy, and QoS in 5G IoT hetnets,” *Applied Sciences (Switzerland)*, vol. 10, no. 20, 2020.

[14] M. Aqib, R. Mehmood, A. Albeshri, and A. Alzahrani, “Disaster management in smart cities by forecasting traffic plan using deep learning and GPUs,” in *Lecture Notes of the Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering, LNICST*, vol. 224. Springer, Cham, nov 2018, pp. 139–154. [Online]. Available: http://link.springer.com/10.1007/978-3-319-94180-6_15

[15] R. Mehmood, S. See, I. Katib, and I. Chlamtac, *Smart Infrastructure and Applications: foundations for smarter cities and societies*, R. Mehmood, S. See, I. Katib, and I. Chlamtac, Eds. Springer International Publishing, Springer Nature Switzerland AG, 2020.

[16] M. A. Sarrayrih and M. Ilyas, “Challenges of online exam, performances and problems for online university exam,” *International Journal of Computer Science Issues (IJCSI)*, vol. 10, no. 1, p. 439, 2013.

[17] A. Ullah, H. Xiao, M. Lilley, and T. Barker, “Privacy and usability of image and text based challenge questions authentication in online examination,” in *2014 International Conference on Education Technologies and Computers (ICETC)*. IEEE, 2014, pp. 24–29.

[18] G. Fenu, M. Marras, and L. Boratto, “A multi-biometric system for continuous student authentication in e-learning platforms,” *Pattern Recognition Letters*, vol. 113, pp. 83–92, 2018.

[19] V. Selvi, R. Sankar, and R. Umarani, “The design and implementation of on-line examination using firewall security,” *IOSR Journal of Computer Engineering*, vol. 16, no. 6, pp. 20–24, 2014.

[20] L. Wei, Z. Cong, and Y. Zhiwei, “Fingerprint based identity authentication for online examination system,” in *2010 Second International Workshop on Education Technology and Computer Science*, vol. 3. IEEE, 2010, pp. 307–310.

[21] K. Garg, K. Verma, K. Patidar, and N. Tejra, “Convolutional neural network based virtual exam controller,” in *2020 4th International Conference on Intelligent Computing and Control Systems (ICICCS)*. IEEE, 2020, pp. 895–899.

[22] Y. Atoum, L. Chen, A. X. Liu, S. D. Hsu, and X. Liu, “Automated online exam proctoring,” *IEEE Transactions on Multimedia*, vol. 19, no. 7, pp. 1609–1624, 2017.

[23] R. Bawarith, D. Abdullah, D. Anas, and P. Dr., “E-exam Cheating Detection System,” *International Journal of Advanced Computer Science and Applications*, vol. 8, no. 4, pp. 176–181, 2017.

[24] Mettl, “Conduct secure online exams with our examination systems,” https://mettl.com/en/computer-based-tests/online-exam-software?utm_source=www.google.com&utm_medium=website&, accessed October 04, 2020.

[25] Proctortrack, “Proctortrack leverages automation and live online remote proctoring using ai to deliver innovation, quality and price leadership at scale.” https://www.proctortrack.com/, accessed October 04, 2020.

[26] Proctoredu, “Conduct credible online tests anywhere, anytime.” https://www.proctoredu.com/, accessed October 04, 2020.

[27] Proctoru, “Protect any online exam,” https://www.proctoru.com/, accessed October 04, 2020.

[28] Comprobo, “Online automated invigilation or proctoring,” https://comprobo.co.uk, accessed October 04, 2020.

[29] N. Dalal and B. Triggs, “Histograms of oriented gradients for human detection,” in *2005 IEEE computer society conference on computer vision and pattern recognition (CVPR’05)*, vol. 1. IEEE, 2005, pp. 886–893.

[30] P. Viola and M. Jones, “Rapid object detection using a boosted cascade of simple features,” in *Proceedings of the 2001 IEEE computer society conference on computer vision and pattern recognition. CVPR 2001*, vol. 1. IEEE, 2001, pp. I–I.

[31] V. Kazemi and J. Sullivan, “One millisecond face alignment with an ensemble of regression trees,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, pp. 1867–1874.

[32] T. Soukupová and J. Cech, “Eye blink detection using facial landmarks,” in *21st computer vision winter workshop, Rimske Toplice, Slovenia*, 2016.

[33] J. Redmon and A. Farhadi, “Yolov3: An incremental improvement. arxiv 2018,” *arXiv preprint arXiv:1804.02767*, pp. 1–6, 2018.