Online Proctoring System

Sri Raaghav. N. U, Aathikrishna. S. K, Anbumani.T, B. Subbulakshmi

Thiagarajar College of Engineering, India

Corresponding author: B. Subbulakshmi, Email: bscse@tce.edu

An online proctoring system will be an effective application to monitor and minimize the malpractices done by students. The main focus here is face detection and face spoofing. Face detection is an artificial intelligence (AI) based computer technology used to find and identify human faces in digital images. It now plays an important role as the first step in many key applications – including face tracking, face analysis, and facial recognition. By using open cv tensor flow libraries we will be building our face detection model. dlib library is used to mark facial pointers from the detected face. We will be also building a face spoofing model to differentiate the devotional phase from images, videos, photos, and stuff. In this model, live video is provided from the integrated camera or web camera as an input to the model so that we would get the result about whether there is any malpractice that has happened or not.

Keywords: Online proctoring, Dlib, Face detection, Facial pointers, Frontal face detection, Haar cascade, Authentication.

2023. In Saroj Hiranwal & Garima Mathur (eds.), *Artificial Intelligence and Communication Technologies*, 137–146. Computing & Intelligent Systems, SCRS, India. https://doi.org/10.52458/978-81-955020-5-9-13

1. Introduction

Face detection is one of the most principal parts of PC vision. It is the foundation of many further examinations from distinguishing explicit individuals to checking central issues on the face. As of late, it has been a considerable amount in the news because of racial profiling occurrences as expounded here where minorities are being misidentified more than white individuals. So significant tech organizations like IBM, Microsoft, and Amazon have restricted their frameworks from being utilized by the police. Multi-task Cascade Convolutional Neural Network (MTCNN),dlibfrontal face detector and DNN module are utilized in this project. Suppose one has applied face lock on their phone and their companion takes it and opens a photo of the user on his phone and shows it and their phone opens. To keep these contemplations from turning into a real face ridiculing strategies are utilized to see whether the face is a genuine individual or just a printed photo or an image of his on an advanced gadget. It is otherwise called enthusiasm identification or replay assault discovery. This paper will cover a strategy given by Costa et al. in his paper, "Picture Based Object Spoofing Detection" whose code they have publicly released too. Likewise, an improvement will be rolled out.

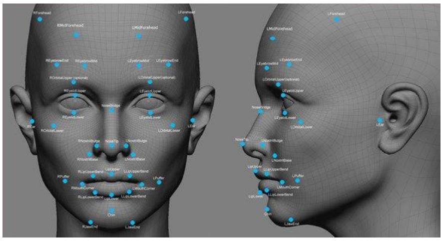


Fig 1. Face marked with facial pointers with the help of dlib

2. Goals

The primary goal is to achieve more accurate face detection and face spoofing.

- To detect malpractice during examination in a faster way as it is processing the live video obtained from the student and not compromising on accuracy.
- To minimize false positives so that it pre-processes our dataset by providing images in YCrCb modes.
  It will be a light-weighted model so that there won't be any processing issues when multiple users access the platform.

3. Literature Survey

Table 1 View of the literature we went through

S.NO	JOURNAL	CONTRIBUTION
1	Online Student Authentication and proctoring system based on multimodel biometric technology. ByMikel Labayen, Ricardo Vea.	In this paper, we describe a new solution based on the authentication of some biometric technologies and an automated proctoring system (system workflow as well as A algorithms), which combines features to solve the main problems in the market: scalability, automatic, affordability, with few hardware and software requirements for the user, reliable and passive for the users. At last, the technological performance test of the system, the usability privacy perception

		survey of the user, and their results are discussed in this work.
2	A Systematic review of online exam solutions in e-learning techniques, tools, and global adoption. by Mohammed Tahir, Abdul Wahab.	Furthermore, 16 important techniques/algorithms and 11 datasets are presented. In addition to this, 21 online exams tools proposed in the selected studies are identified. Additionally, 25 leading existing tools used in the selected studies are also presented. Finally, the participation of countries in online exam research is investigated. Key factors for the global adoption of online exams are identified and compared with major online exams features. This facilitates the selection of the right online exam system for a particular country based on the existing E-learning infrastructure and overall cost. To conclude, the findings of this article provide a solid platform for the researchers and practitioners of the domain to select appropriate features along with underlying development approaches, tools, and techniques for the implementation of a particular online exams solution as per the given requirements
3	Integrated deep model for face detection and landmark localization. -by Gary storey, Ahmed Bouridane. .	The tasks of face detection and landmark localization are a key foundation for many facial recognition analysis applications, while great advancements have been achieved in recent years there are still challenges to increasing the precision of face detection. Within this paper, we present our novel method the Integration Deep Model (IDM)
4	A face Spoofing detection method based on domain adaptation and lossless size adaptation. -by Wenyun Sun,Yu Song,Zhong Jin.	In this paper, a face spoofing detection method called the Fully Convolutional Network with Domain Adaptation and Lossless Size Adaptation (FCN-DA-LSA) is proposed. As its name suggests, the FCN-DA-LSA includes a lossless size adaptation preprocessor followed by an FCN based pixel-level classifier embedded with a domain adaptation layer. The FCN local classified hakes full use of the basic properties of face spoof distortion namely ubiquitous and repetitive. The domain adaptation (DA) layer improves generalization across different domains.
5	A Cascade face spoofing detector based on Face Anti-Spoofing R-CNN and improved Retinex LBP. by Haonan Chen,Yaowu Chen,Xiang Tian.	In this paper, we design a face anti-spoofing region-based convolutional neural network (FARCNN). based on the improved Faster-region-based convolutional neural network (R-CNN) framework. Motivated by face detection, we regard face spoofing detection as a three-way classification to distinguish a real face. fake face and background. We extend the typical Faster R-CNN scheme by optimizing several important strategies, including ROI-pooling feature fusion and adding the Crystal Loss function to the original multi-task loss function.

4. Face Detection

Face detection is one of the essential factors of pc vision. It is the base of many similar kinds of research like figuring out particular humans to mark key factors on the face. Recently, it's been a pretty lot with inside the information because of racial profiling incidents as elaborated wherein humans of color are being misidentified greater than white humans. So essential tech groups like IBM, Microsoft, and

Amazon have banned their structures from being utilized by the police. However, this text won't be living on one's factors and we can simply be seeking to draw bounding containers on faces with the usage of pre-skilled fashions like Haar cascades, dlib frontal face detector, MTCNN, and a Caffe version the usage of OpenCV's DNN module. Then it will be evaluated to discover which matches the exceptional for real-time applications.

4.1. Face Detection Models

Haar Cascades

They were proposed again in 2001 by Paul Viola and Michael Jones in their paper, "Rapid Object Detection the use of a Boosted Cascade of Simple Features." It is incredibly rapid to work with and just like the simple CNN, it extracts a whole lot of features from images. The best features are then decided through Adaboost. This reduces the authentic 160000+ capabilities to 6000 capabilities. But making use of these types of features in a sliding window will nevertheless take a whole lot of time. So, they added a Cascade of Classifiers, wherein the features are grouped. If a window fails at the primary stage, those remaining features in that cascade aren't processed. If it passes then the following characteristic is examined and the same procedure is repeated. If a window can pass all the features then it is classified as a face region.Haar cascades require a lot of positive and negative training images to train. Thankfully, these cascades come bundled with the OpenCV library along with the trained XML files.

Dlib frontal face detector

Dlib is a C++ toolkit containing machine learning algorithms used to solve real-world problems. Though it's written in C++ it has python bindings to run it in python. It additionally has a nice facial landmark keypoint detector. The frontal face detector provided by dlib works using features extracted by Histogram of Oriented Gradients (HOG) which are then passed through an SVM. In the HOG feature descriptor, the distribution of the directions of gradients is used as a feature.

MTCNN

It was introduced by Kaipeng Zhang, et al. in 2016 in their paper, "Joint Face Detection and Alignment victimization Multi-task Cascaded Convolutional Networks." It not only detects the face but additionally detects 5 key points as well. It uses a cascade structure with 3 stages of CNN. First, they use a fully convolutional network to get candidate windows and their bounding box regression vectors, and the highly overlapped candidates are overlapped using on-maximum suppression (NMS). Next, these candidates are passed to a different CNN that rejects an outsized variety of false positives and performs the activity of bounding boxes. In the final stage, facial landmark detection is performed.

DNN

It is a Caffe model that relies on the Single Shot-Multibox Detector (SSD) and uses ResNet-10 architecture as its backbone. It was introduced post OpenCV 3.3 in its deep neural network module. There's conjointly an amount of TensorFlow version that may be used however we'll use the Caffe Model.

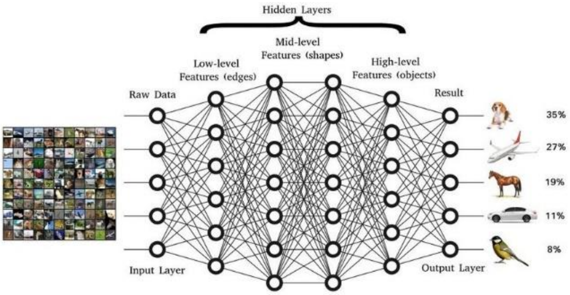

Hidden Layers
Mid-level Features (shapes)
Low-level Features (edges)
High-level Features (objects)
Raw Data
Result
Input Layer
Output Layer
35%
27%
19%
11%
8%

Fig.2. Layers of DNN

4.2. Facial Point Detectors

The initial step is to discover the appearances in the pictures on which we can get facial milestones. This assignment will utilize a Caffe model of OpenCV's DNN module. In case one is thinking about how it performs against different models like Haar Cascades or Dlib's frontal face detector, We will utilize a facial milestone indicator given by Yin Guobing. It additionally gives 68 tourist spots and it is a Tensorflow CNN prepared on 5 datasets.

The current arrangements like OpenFace and Dlib's facial milestone discovery alongside the datasets are accessible. It is needed to do information preprocessing and preparing it to utilize, then, at that point, it is required to remove the appearances and apply facial milestones on it to prepare a CNN and store them as TFRecord documents. Then, at that point, a model is prepared to utilize Tensorflow.

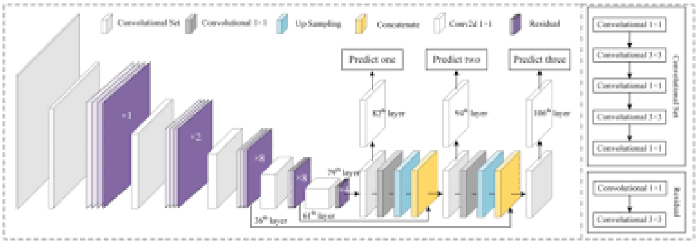


Fig 3. This diagram represents the YLOV3 model. Source: stackoverflow.com

Eventually, the model is traded as an API and used in Python. We need the directions of the appearances that go about as the area of interest and are extricated from the picture. Then, at that point, it is changed over to a square state of size 128x128 and passed to the model which returns the 68 central issues which would then be able to be standardized to the elements of the first picture.

5. Face Spoofing

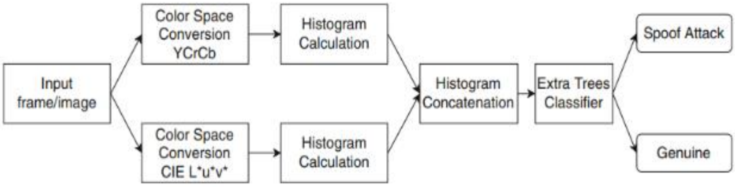


Fig 4: Face Spoofing

The methodology here is to utilize distinctive colorspaces like YCrCb and CIE L*u*v* and make histograms in these channels. Normal RGB colorspace isn't utilized because the connection between the red, green, and blue channels implies that it discourages detachment among luminance and chrominance which is fundamental in spoofing attacks. In the paper, they talked about the mathematics of changing RGB over to YCrCb and CIE L*u*v*, notwithstanding, we don't have to do that here as there are as of now predefined works in OpenCV to do that. The histograms are connected into an element vector FV = (Y, Cr, Cb, L, u, v) of size 1536 that is passed as an input to an additional tree classifier for training purposes. They tried their strategy on faces as well as on genuine and printed plug plugs of wine bottles.

Requirements

To use the pre-trained model, we require Sklearn version 0.19.1 as the model was trained using it and due to some depreciations, the current version 0.23.1 does not run it. Other than that we would need OpenCV to use the webcam or handle images and for changing the color channels.

6. Analysis

- Haar Cascade provides comparably high false positives than other modules.
- Dlib and MTCNN had very similar results but MTCNN has slightly improved results and dlib can't identify very small faces. If the images are large, then there is a high possibility that the lighting will be good along with low occlusion, and mainly front-facing faces MYCNN may give the best results.
- For general problems, the Caffe model of OpenCV provides the best results. Compared to other models, DNN has the highest frame rate.
- One of the significant selling points of Dlib was its speed. How about we perceive how they analyze on the i5 processor.
- Dlib gives ~11.5 FPS and the milestone forecast step takes around 0.005 seconds.
- The Tensorflow model gives ~7.2 FPS and the milestone forecast step takes around 0.05 seconds.
- So in case of speed is the primary concern and impeded or calculated countenances less then Dlib may be more qualified for us in any case we feel that the Tensorflow model rules without compromising a great deal on speed.
- The face region is removed and the changed over to YCrCb and Luv from BGR (OpenCV utilizes BGR rather than RGB) and afterward, histograms are determined, trailed by their connection to make a component vector which we will pass to our additional trees model. At last, then, at that

point, we foresee the likelihood of whether it is certifiable or phony and the limit for genuineness is set to 0.7.

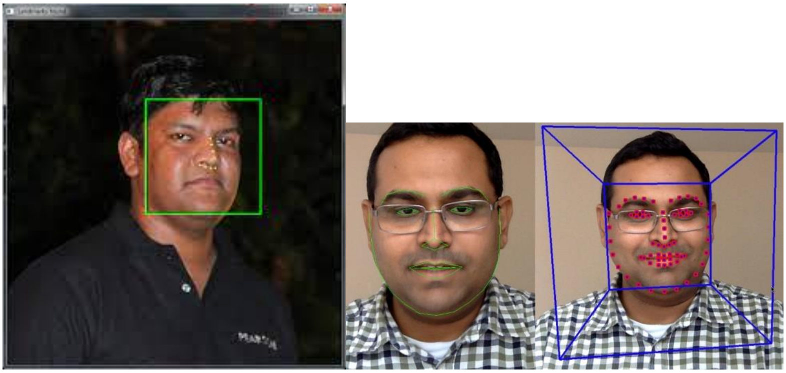
If the result to be obtained is of higher accuracy, it is recommended to use Faster-RCNN. If we are ready to compromise on accuracy but want higher processing speed, then it is recommended to use YOLOv3. If we want a trade-off between the above-mentioned models then it is recommended to use SSD. Here, the scenario is that there will be multiple users logging into the site simultaneously, so the processing speed should be high to manage this scenario properly. So, we are using the YOLOv3 model.

7. Dataset

The primary aim here is to keep the model light weighted. So, we are going to use pre-trained models from existing libraries. To increase the performance, it is required to pre-process the data effectively in the view of reducing the possibility of false positives. The dataset used here is the live video captured from the user during the exam through an integrated camera or web camera.

8. Results

Here, DNN is used which is an object detection model. Here, the face of the user is recognized as an object by the model, and that recognized area is shown with a rectangular box. This recognized area is then sent to Dlib which has the feature of facial point detection. This library is used to plot 68 points all around the recognized area and if all the 68 pointers are plotted correctly, then the recognized area can be concluded as a human face.



The approach of face spoofing is to use different color spaces such as YCrCb and CIE L*u*v* to create histograms. Luminance and chrominance are very much essential for spoofing attacks. So, Normal RGB color spaces cannot be used as they obstruct the correlation between red, green, and blue channels. The obtained histograms are then converted into feature vectors which can be passed as input for training purposes.

9. Conclusion

Thus, the model is now ready with facial pointers with the help of facial point detection. Using these facial pointers, the methodology of eye tracking can be implemented by using the facial pointers around the eyeball. Thus, the eyeball movements can be tracked and a threshold is set. If the movement exceeds the threshold, then a notification is sent to the invigilator about the possibility of malpractice. Then, the facial movements can also be detected concerning the relative angles between the facial pointers. Here also, we set a threshold and if the angle exceeds the threshold, then it is taken as an attempt of malpractice. Also, we can monitor the mouth opening and closing of the user and if there exists any attempt and the relative distance between certain pointers then it is taken as an attempt of malpractice. The output is a Boolean value that will be sent to the invigilator so that he/she will be notified whether the student has committed malpractice or not.

10. Future Enhancement

- Make a better face spoofing model as the accuracy may not be up to the mark currently.
- Use a smaller and faster model in place of YOLOv3 that can give good FPS on a CPU.
- Add a vision-based functionality: face recognition such that no one else replaces the candidate and leaves the exam midway.
- Add a vision-based functionality: id-card verification.

References

[1] J. Li, Y. Wang, T. Tan, and A. K. Jain, "Live face detection based on the analysis of Fourier spectra," Proc. SPIE, vol. 5404, pp. 296–304, Aug. 2004.

[2] X. Tan, Y. Li, J. Liu, and L. Jiang, "Face liveness detection from a single image with a sparse low-rank bilinear discriminative model," in Proc. Eur. Conf. Comput. Vis. Berlin, Germany: Springer, 2010, pp. 504–517.

[3] J. Määttä, A. Hadid, and M. Pietikäinen, "Face spoofing detection from single images using micro-texture analysis," in Proc. Int. Joint Conf. Biometrics (IJCB), Oct. 2011, pp. 1–7.

[4] T. de Freitas Pereira, A. Anjos, J. M. De Martino, and S. Marcel, "LBP – TOP based countermeasure against face spoofing attacks," in Proc. Asian Conf. Comput. Vis. Berlin, Germany: Springer, 2012, pp. 121–132.

[5] D. Gragnaniello, G. Poggi, C. Sansone, and L. Verdoliva, "An investigation of local descriptors for biometric spoofing detection," IEEE Trans. Inf. Forensics Security, vol. 10, no. 4, pp. 849–863, Apr. 2015

[6] F. Deeba, H. Memon, F. Ali, A. Ahmed, and A. Ghaffar, "LBPHbasedenhanced real-time face recognition," Int. J. Adv. Comput. Sci. Appl., vol. 10, no. 5, pp. 274–280, 2019.

[7] K. Arya, S. S. Rajput, and S. Upadhyay, "Noise-robust low-resolution face recognition using sift features," in Computational Intelligence: Theories, Applications, and Future Directions, vol. 2. New York, NY, USA: Springer, 2019, pp. 645–655.

[8] A. Ahmed, J. Guo, F. Ali, F. Deeba, and A. Ahmed, "LBPH based improved face recognition at low resolution," in Proc. Int. Conf. Artif. Intell. Big Data (ICAIBD), May 2018, pp. 144–147.

[9] D. Menotti, G. Chiachia, A. Pinto, W. Robson Schwartz, H. Pedrini, A. Xavier Falcao, and A. Rocha, "Deep representations for iris, face, and fingerprint spoofing detection," IEEE Trans. Inf. Forensics Security, vol. 10, no. 4, pp. 864–879, Apr. 2015.

[10] Y. A. Ur Rehman, L. M. Po, and M. Liu, "Deep learning for face antispoofing: An end-to-end approach," in Proc. Signal Process., Algorithms, Archit., Arrangements, Appl. (SPA), Sep. 2017, pp. 195–200.

[11] C. Nagpal and S. R. Dubey, "A performance evaluation of convolutional neural networks for face anti-spoofing," in Proc. Int. Joint Conf. Neural Netw. (IJCNN), Jul. 2019, pp. 1–8

[12] Mashat, A. F., Fouad, M. M., Philip, S. Y., & Gharib, T. F. (2012). “A decision tree classification model for university admission system”. Editorial Preface, 3(10).

[13] A. Natawiguna and M. M. I. Liem, “Virtualization methods for securing online exam,” in Proc. Int. Conf. Data Softw. Eng. (ICoDSE), Oct. 2016, pp. 1–7.

[14] M. Ghizlane, B. Hicham, and F. H. Reda, “A new model of automatic and continuous online exam monitoring,” in Proc. Int. Conf. Syst. Collaboration Big Data, Internet Things Secur. (SysCoBIoTS), Dec. 2019, pp. 1–5.

[15] N. Ketui, K. Homjun, K. Poonyasiri, J. Deepinjai, and P. Luekhong, “Itembased approach for online exam performance and its application,” in Proc. 13th Int. Conf. Electr. Eng./Electron., Comput., Telecommun. Inf. Technol. (ECTI-CON), Jun. 2016, pp. 1–5.

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

[17] A. A. Sukmandhani and I. Sutedja, “Face recognition method for online exams,” in Proc. Int. Conf. Inf. Manage. Technol. (ICIMTech), Aug. 2019, pp. 175–179.

[18] M. Cote, F. Jean, A. B. Albu, and D. Capson, “Video summarization for remote invigilation of online exams,” in Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV), Mar. 2016, pp. 1–9.

[19] G. Sukadarmika, R. S. Hartati, Linawati, and N. P. Sastra, “Introducing TAMEx model for availability of e-exam in wireless environment,” in Proc. Int. Conf. Inf. Commun. Technol. (ICOIACT), Mar. 2018, pp. 163–167.

[20] M. Valstar, B. Martinez, X. Binefa, and M. Pantic, “Facial point detection using boosted regression and graph models,” in Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit., Jun. 2010, pp. 2729–2736.

[21] S. Ren, X. Cao, Y. Wei, and J. Sun, “Face alignment at 3000 FPS via regressing local binary features,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., Jun. 2014, pp. 1685–1692.

[22] A. Bulat and G. Tzimiropoulos. (Sep. 2017). “How far are we from solving the 2D & 3D Face Alignment problem? (and a dataset of 230,000 3D facial landmarks).” [Online]. Available: https://arxiv.org/abs/1703.07332

[23] M. Jones and P. Viola, “Fast multi-view face detection,” Mitsubishi Electr. Res. Lab., Cambridge, MA, USA, Tech. Rep. TR20003-96, Jul. 2003.

[24] P. J. Phillips, H. Moon, S. A. Rizvi, and P. J. Rauss, “The FERET evaluation methodology for face-recognition algorithms,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 22, no. 10, pp. 1090–1104, Oct. 2000.

[25] B. Heisele, T. Serre, and T. Poggio, “A component-based framework for face detection and identification,” Int. J. Comput. Vis., vol. 74, no. 2, pp. 167–181, 2007

[26] S. Sawhney, K. Kacker, S. Jain, S. N. Singh, and R. Garg, “Real-time smart attendance system using face recognition techniques,” in Proc. 9th Int. Conf. Cloud Comput., Data Sci. Eng. (Confluence), Jan. 2019, pp. 522–525.

[27] A. Alshbtat, N. Zanoon, and M. Alfraheed, “A novel secure fingerprintbased authentication system for student’s examination system,” Int. J. Adv. Comput. Sci. Appl., vol. 10, no. 9, pp. 515–519, 2019. [Online]. Available: https://thesai.org/Publications/ViewPaper?Volume=10&Issue=9&Code= IJACSA&SerialNo=68.

[28] J. V. Monaco, J. C. Stewart, S.-H. Cha, and C. C. Tappert, “Behavioral biometric verification of student identity in online course assessment and authentication of authors in literary works,” in Proc. IEEE 6th Int. Conf. Biometrics, Appl. Syst. (BTAS), Sep. 2013, pp. 1–8.

[29] E. Flior and K. Kowalski, “Continuous biometric user authentication in online examinations,” in Proc. Int. Conf. Inf. Technol., Jan. 2010, pp. 488–492.

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

[31] A. Okada, I. Noguera, L. Alexieva, A. Rozeva, S. Kocdar, F. Brouns, T. Ladonlahti, D. Whitelock, and A. Guerrero-Roldán, "Pedagogical approaches for e-assessment with authentication and authorship verification in higher education," Brit. J. Educ. Technol., vol. 50, no. 6, pp. 3264–3282, Nov. 2019.

[32] G. Fenu, M. Marras, and L. Boratto, "A multi-biometric system for continuous student authentication in e-learning platforms," Pattern Recognit. Lett., vol. 113, pp. 83–92, Oct. 2018.

[33] L. Slusky, "Cybersecurity of online proctoring systems," J. Int. Technol. Inf. Manage., vol. 29, no. 3, pp. 56–83, 2020