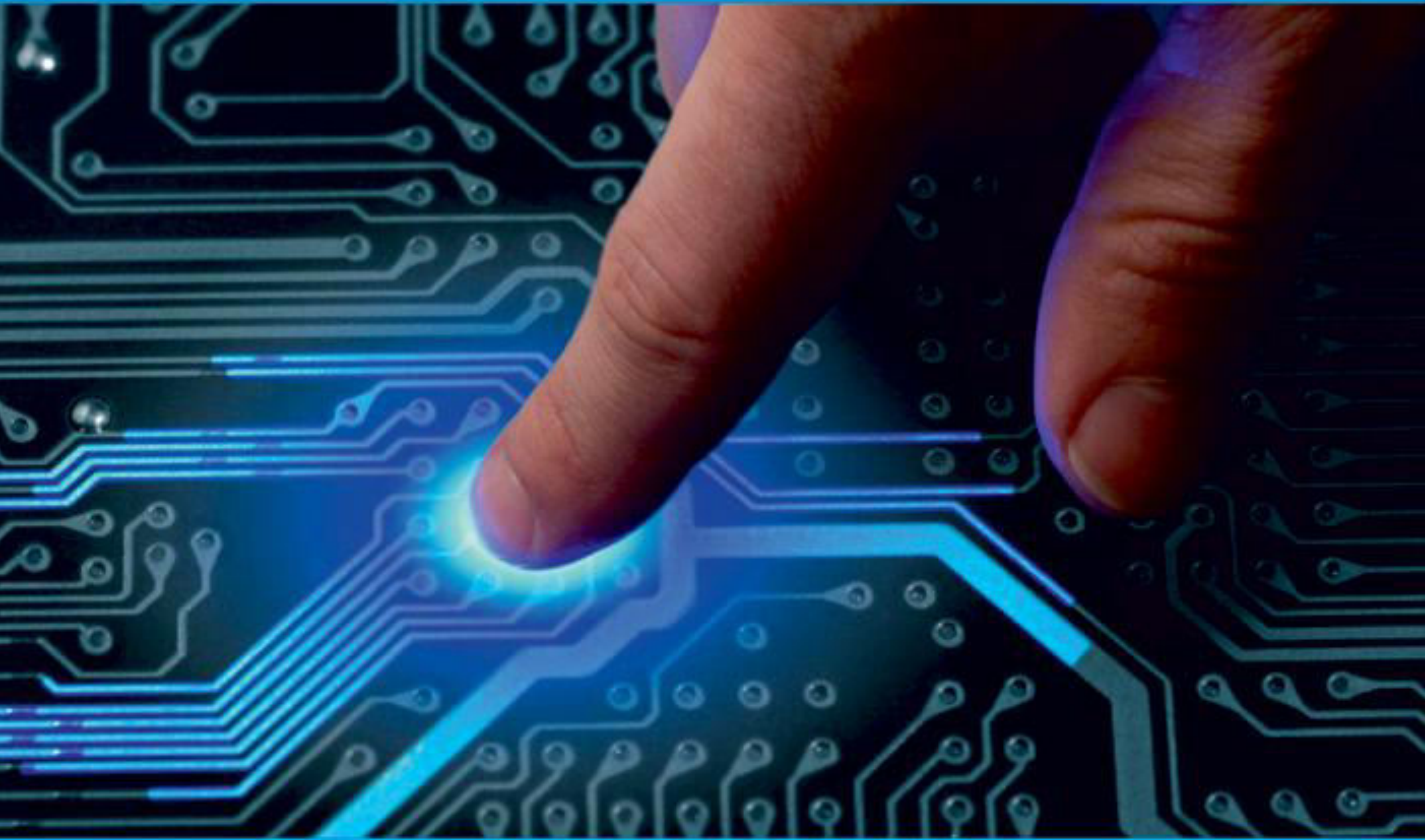
IJIRCCE

e-ISSN: 2320-9801 | p-ISSN: 2320-9798



INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN COMPUTER & COMMUNICATION ENGINEERING

Volume 9, Issue 7, July 2021

INTERNATIONAL STANDARD SERIAL NUMBER INDIA

Impact Factor: 7.542

International Journal of Innovative Research in Computer and Communication Engineering

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 7.542

|| Volume 9, Issue 7, July 2021 ||

| DOI: 10.15680/IJIRCCE.2021.0907144 |

AI Based Proctoring System

Nikitha K S, Abhilash Kumar Singh, Harsha C, Harshith Gowda N, Hari L M

Assistant Professor, Dept. of CS&E, Bangalore Institute of Technology, VV Puram, K R Pura, Bangalore India

UG Student, Dept. of CS&E, Bangalore Institute of Technology, VV Puram, K R Pura, Bangalore India

UG Student, Dept. of CS&E, Bangalore Institute of Technology, VV Puram, K R Pura, Bangalore India

UG Student, Dept. of CS&E, Bangalore Institute of Technology, VV Puram, K R Pura, Bangalore India

UG Student, Dept. of CS&E, Bangalore Institute of Technology, VV Puram, K R Pura, Bangalore India

ABSTRACT: The covid pandemic has affected the educational system worldwide, leading to the near-total closures of schools, universities and colleges. We were not ready for this on the academic front. Tests were conducted online. Precisely, there merely exists any system that conducts institutional tests in a proctored way. Lately the tests were conducted via meeting applications such as Google meet, Microsoft teams, zoom etc., where a single proctor has to watch over multiple students simultaneously. This approach is highly inefficient and no- trustworthy. We came across the idea of building such an application that rules out all the above disadvantages. An application that is very user-friendly, has some automation and is technically very rich to conduct online exams. In its first iteration, features like facial recognition, audio detection, gaze detection, object detection and advanced behaviour detections are present. Currently, accuracy expected is more than 80%, and to prevent false positive or false negative prediction, we will also provide evidence of malpractice. Algorithms will be self-learning which will help our application to gain more accuracy over the years. This application will most likely cross all its barriers as it can be used by multiple colleges simultaneously and effectively.

KEYWORDS: Face recognition; Face detection; Object detection; Face direction detection; Eye gaze tracking

I. INTRODUCTION

The global pandemic has affected all parts of our lives making us take many measures to prevent exposure to said virus. Educational Institutions, ranging from nursery to Degree colleges could no longer function as they used to, they had to conduct classes online. The exams/assessments conducted by universities were switched to online modes, although this was an effective way for the students to attend the exams, but it was not effective when the teachers had to monitor for malpractices, because a single teacher cannot monitor 20+ students at the same time, and the examinees are incredibly sharp in discovering ways to commit malpractice. In order to combat the mentioned issues, we will develop a software system which will monitor all the students concurrently and detect as much malpractice as possible, by using Artificial Intelligence techniques, such as facial recognition, facial detection, object detection, voice activity detection etc. In university exams, lakhs of students will be writing the exam simultaneously, due to which there will be heavy load on the server. We have designed architecture which is highly scalable. Moreover, we tried to keep a good balance between speed and accuracy.

II. RELATED WORK

Heuristic Based automatic online proctoring system by Vishnu Raj,et al, [1] They proposed as an initial checking process, the student will have to show the room. Video and audio of the candidate will be examined. To eliminate the possibility that candidates might open files and copy, they are monitoring active window details. Face tracking. They had extreme proctoring ideas, if face is missing at any point of time, they conclude malpractice. Sound analysis was not sufficient.

An intelligent system for online exam monitoring by Swathi prathish,et al, [2] In their system, they record the video of the candidate throughout the exam and at the end the video was broken into frames. Face detection and landmark localization was performed. Basic audio monitoring and tab switch monitoring. Drawbacks observed are inappropriate audio monitoring, sensitive to less sound also. No pc monitoring feature. At the end of the exam students have to wait for a couple of minutes while the above processing takes place.

There exist many applications in the market to conduct online competitive examination such as, hackerearth, hackerrank, mettle, hirepro.

III. OBJECTIVES

- To Collect COVID-19 data and store in a suitable form.
- To pre-process and extract useful features from data.
- To train a Machine Learning Model on these extracted features.
- To Classify the data-point as an Outlier/Non-Outlier.
- To evaluate model-performance and improve it.
- To perform hyper parameter tuning on components of model.

IV. PROPOSED SYSTEM

Developing a website for to teachers to create exam, invite student, examine the test scripts.

- Developing a desktop application where student can write his/her exam.
- Developing efficient cheat proof AI proctored system with the below features
  - Facial detection and recognition of the examinee
  - Object detection i.e., phone detection etc.
  - Multiple people detection
  - Eye direction and head motion detection
  - Plagiarism detection
- To detect when someone moves away from the examination window
- To send proof to the examinee if their exam is rejected due to misconduct
- To provide a fast system with minimum delay through every step

V. PROJECT MODULES

A. Flask Server

We have hosted a flask server (Python) on Heroku which provides an interface to create and manage exams. It also provides protected services to the other counterpart of this module i.e., the electron desktop application

- *Authentication Module* This module deals with authentication from two sources. It provides a login interface where a proctor can use his username and password to login to it. It provides an interface where an examinee can authenticate himself by providing an authentication token created using HMAC256 Algorithm
- *Proctor Module* This contains all the routes provided to the proctor and the administrator.
- *Exam API Module* the API is protected by HTTP Token authentication method. It provides things such as question bank retrieval, dynamic answer update, etc.
- *A3H Module* This module has access to the list of all the logs generated by users, and trains an ML model on it and then it can be used to predict the A3H Score for new student exam logs.

B. Exam Desktop Application

- *Login Module* This Module provides a means of authenticating the students and their access to the proctor server.
- *Environment Configuration Module* It mainly checks if the required dependencies are installed on the host computer, and some preliminary model training
- *Proctored Examination Module* It handles the examination view, proctoring, face recognition, and also dynamic answer updates.
- *Logger Module* This module is responsible for logging important events from all the other modules. The events may be face not recognized, activity found in microphone, or user disconnected. After Exam, the logs are sent into the proctor server to be analyzed by the A3H algorithms.

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 7.542

|| Volume 9, Issue 7, July 2021 ||

| DOI: 10.15680/IJIRCCE.2021.0907144 |

V. SYSTEM ARCHITECTURE

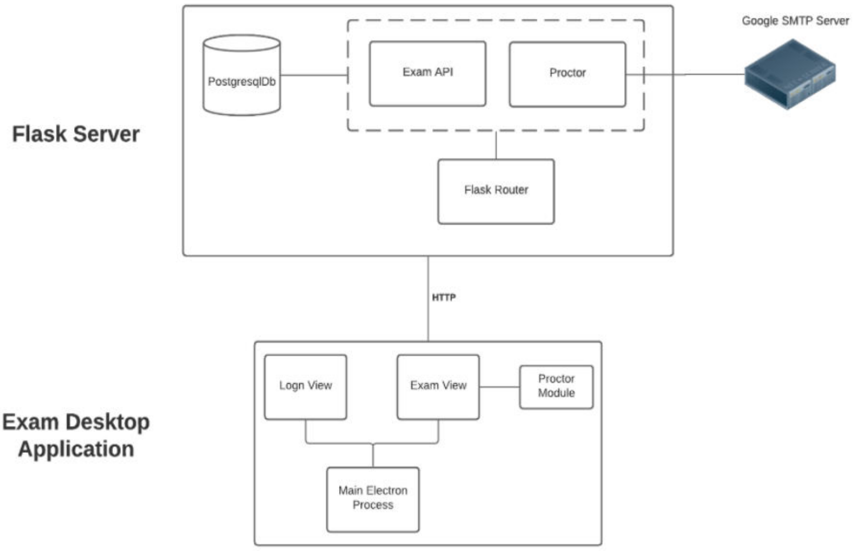


Figure-1 System architecture

The above diagram shows the basic architecture of our system. Flask web application is for the teachers to create exam, add and invite students and correct the answer script. Whereas, we have built a desktop application for students to attempt the exam. This desktop application will be communicating with flask server to save the answer script and proctoring logs. All the proctoring modules will be running on user's computer independently, without server interaction, because of this lot of load is shifted to the client side from server side.

Auto-saving feature is implemented for answers in desktop application. Moreover, desktop application runs on full screen mode, so there is no way by which examinee can minimize and use search engine to cheat. In case of power failure, all the answers will be stored in local database, until the power comes back.

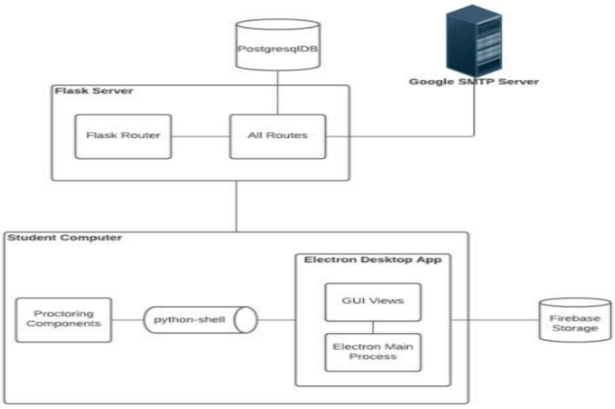


Figure-2 Component-to-component interaction

The above diagram describes the interaction between components of our system, the components belong to two main modules, the backend application and the student desktop application. Flask router component route HTTP requests to one of the request handlers in All Routes Component, the backend also has access to a Google's smtp server because it has to send emails to students. The exam desktop application is governed by the main electron process, which spawns Gui windows using Browser Windows Instances, and it also can access proctoring components written in python using the python shell interface to manage them. Electron also has an interface to firebase storage for file uploads.

We have used res10 pretrained model for face detection, open face for face recognition. Res50 for custom object detection, frontal face and 68_landmark.Dat for landmarking.

VI. ALGORITHM

Algorithm 1: Proctoring Backend

```
Input: Video frames and audio
Output: A3H score
Loop (time < endTime)
        Take video frame
        Apply facial detection
        If (No_of_face detected>1) Then Multiple_face_counter+=1
        Else Ignore
                Take 10 sec audio input for monitoring, apply speech_recognition and NLP
        If (suspicious) Then Audio_flag+=1
        Else Ignore
                Apply object detection
        If (suspicious) Then object_detected+=1
        Else Ignore
                Apply Facial Recognition, motion detection
        If (suspicious) Then increase the counter for respective field
        Apply plagiarism check among answers of students
        Update the degree of plagiarism
Test the evaluated data on A3H dataset. It returns the degree of cheating
```

Algorithm 2: FACE TRAINING

```
t: Video stream to capture 1400 images of user
ut: Pickle file containing face embedding and name of person
  Start
  Initialise Video object and start capturing video stream
  Initialise frame_count=0
  While True:
          Save each frame as a png image
          If frame_count>=1400:
                  Stop the video stream
                  Release the camera object
          Load face detection model,i.e. , caffe model and prototype text
          Convert caffe model and network model
          Load face recognition model, i.e., torch model
          Convert torch to network model
          Traverse over face database and store the path to 'filenames' list
          Get the image file and perform image preprocessing into blob
          Pass this blob as an input to detection network and perform forward propagation
          Loop over the detection
                  Extract confidence
                  If Confidence > 0.50 then
                          Extract the coordinates of face, crop the face from image
                          Perform preprocess of face
                          Pass this face to recognition model and perform forward propagation
                  Get the face embeddings as an output of this model, store embeddings in a list
abelEncoder on name list
e embedding and labelEncoded name to SVC model
p the SVC trained model in recognize pickle file
p the labelEncoder in le pickle file
  Save the logs in configuration log text file
```

Algorithm 3: FACE RECOGNITION

```
t: Video stream from camera
ut: Proctoring logs in text file, i.e., known face, unknown face, no face or multiple face

 face detection model,i.e. , caffe model and prototype text
ert caffe model to network model
 face recognition model, i.e., torch model
ert torch to network model
:kle recognizer and le pickle file which contains the face embedding and name of person
video stream and initialise count=0
e True:
        If count >= 100, stop the video stream, and release camera object
        Fetch the frame, preprocess and normalize the frame and convert into blob
        Supply this blob to network and perform forward propagation
        Loop over the detection
          Extract confidence
          If confidence>0.50:
                  Extract the coordinates of face, crop the face from image
                  Perform preprocess of face
                  Pass this face to recognition model and perform forward propagation
                  Get the face embeddings as an output of this model
>are this face embedding with unpickled recognizer embeddings
tch:
                  Get the name of person from le pickle file
                  Open a text file in append log
                  Write the log of no. of unknown people and known person
                  if no. Of person > 1:
the frame in png/jpg format
  Close file
onfidence < 0.50: log it in txt file as no face detected and save the frame in jpg format
ase the frame count, i.e., count+=1
  Release the camera object and stop streaming
```

Algorithm 3: FACE MOTION

```
Input: Video stream
Output: x-axis, y-axis deviation values
Import dlib and load "shape_predictor_68_face_landmarks.dat" model
Initialise camera object and start video stream
While True:
        Get the frame from the video stream
        Convert the image to grayscale
        Use model detector object to detect 68 landmarks on the face
        Crop the left and right cheek portion using the cheeks landmark by applying mask to gray image
        Compare the white pixel in left and right cheek
Release the camera object
```

VII. PSEUDO CODE

- Load res10_ssd_300x300 model
- Load openface_nn4_small model
- Load user recognizer
- Set video device to /dev/video0
- While Exam Runs
  - Capture a frame from video device
  - Get bounding box of face from frame

        - If multiple face exists or face not detected
            - Write to logs and restart loop
        - Recognize face from photo
        - If unknown face recognized
            - Write to logs and restart
        - Detect face alignment
        - Detect eye position
        - Write to logs
    - Release loaded device
    - Commit all logs to log file

VIII. RESULTS

Manual test was performed on various unit component of the system. Current frame rate is 3 frames per second since, we have used bulky modules. Facial recognition is very efficient, since out of 100 tests performed, we got 1 false positive. To train 1400 positive and 2000+, instances of face, it takes around 3 mins of time. Since all the cheating evidence is stored on the user side, there is no issue of privacy. Currently frontal face model is bit inaccurate, since this model fails, when we turn faces 45 degrees on either side, future enhancements will be done on it. Logs are successfully captured by the server, and both the A3H score and cumulative logs were shown to the teacher. Face motion and eye gaze is also near real time.

IX. CONCLUSION, APPLICATIONS AND FUTURE WORK

A. *Conclusion*

The proposed system tries to automate the proctoring of university examination to battle the current remote proctoring scenario and increase the quality of learning and better user experience.

B. *Applications*

- Cheating proof examination for university and school exams as well as for recruitment.
- Paperless online examination.
- Advanced grading system i.e., auto check functionality for examiners.

C. *Future Scope of the Project*

- More sophisticated urban sound elimination from audio proctoring.
- Digital ID CARDS
- Better GUI to create diagrams in desktop application.

REFERENCES

[1] R.S. Vishnu Raj, S. Narayana and Kamal Bijlani, "Heuristic based automatic online proctoring system", 2015 IEEE 15th International Conference on Advanced Learning Technologies.

[2] Prathish, S., Athi Narayanan S, & Bijlani, K. (2016). An intelligent system for online exam monitoring. 2016 International Conference on Information Science (ICIS).

[3] Tyas Purwa Hapsari, D., Gusti Berliana, C., Winda, P., & Arief Soeleman, M. (2018). Face Detection Using Haar Cascade in Difference Illumination. 2018 International Seminar on Application for Technology of Information and Communication.

[4] Parkhi, Omkar & Vedaldi, Andrea & Zisserman, Andrew. (2015). "Deep Face Recognition".

[5] TCS NQT, "TCS NQT online examination rules", circulated via mails to the candidates.

[6] Helaine M. Alessio, Nancy Malay, Karsten Maurer, A. John Bailer, and Beth Rubin, (2017) "Examining the Effect of Proctoring on Online Test Scores" in Miami University, Online Learning 21 (1).

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN COMPUTER & COMMUNICATION ENGINEERING