Design and Development of a Cutting-Edge Machine Learning-Driven Virtual Learning Platform to Revolutionize Online Education and Improve Student Learning during COVID-19

Mavis Malachi Ejiofor [1], **Taiwo Abdulahi Akintayo** [2], **Agbonze Nosa Godwin** [3]

[1] *Lagos State University*
Lasu Main Rd, Ojo, Lagos 102101, Lagos State, Nigeria

[2] *National Centre for Artificial Intelligence and Robotics*
No 28, Port Harcourt Crescent, Off Gimbiya Street, P. M. B 564, Area 11, Garki, Abuja, Nigeria

[3] *University of Benin*
P. M. B. 1154, Ugbowo, Benin – Ore Rd, Edo State, Nigeria

DOI: 10.22178/pos.106-33

LCC Subject Category: S1-(972)

Received 19.06.2024
Accepted 28.07.2024
Published online 31.07.2024

Corresponding Author:
Mavis Malachi Ejiofor
Ejioformavis@gmail.com

Abstract. Instructors in virtual classes are facing previously unheard-of difficulties in sustaining student engagement and attendance as the COVID-19 pandemic continues to alter the education landscape. To solve this pressing problem, we have created facial analysis technology that enables teachers to track students' engagement and attention in real-time.

Our user-friendly platform uses cutting-edge face detection technology and machine learning to give teachers a visual dashboard that shows disengaged students as red boxes and engaged students as green boxes. This cutting-edge tool helps teachers determine which students need more encouragement or support, guaranteeing individualized attention and better learning results.

Our tool provides instructors with features, such as automated attendance records and early departure detection, that go beyond simple attendance tracking and help them optimize online class management. Our solution seeks to humanize online learning by utilizing facial analysis to provide students with a more engaging and productive learning environment.

Keywords: Facial analysis; Python; Machine learning; student engagement; instructor support; virtual classroom; COVID-19.

INTRODUCTION

The COVID-19 pandemic has transformed the education sector, necessitating innovative solutions for remote learning [1, 2]. Classroom management software has emerged as a vital tool, streamlining administrative tasks, enhancing organization, and facilitating seamless student-teacher connections [3, 4]. This software is a central hub, fostering collaborative ecosystems for academic excellence [1]. Teachers can focus on delivering high-quality instruction and promoting student growth by automating tasks, providing immediate feedback, and offering personalized learning pathways [2].

Effective classroom management software is the foundation of a successful and productive learning environment [1]. It is a central hub, bringing educators, learners, and parents together in a collaborative ecosystem that fosters academic excellence [2]. By automating repetitive tasks [3], giving educators immediate feedback [4], and offering personalized learning pathways [5], this software frees up teachers to concentrate on what matters—delivering high-quality instruction and fostering student growth [6]. Nevertheless, the advantages of classroom management software go well beyond the classroom, enhancing student engagement [7], facilitating online learning [8], and ensuring security and privacy [9].

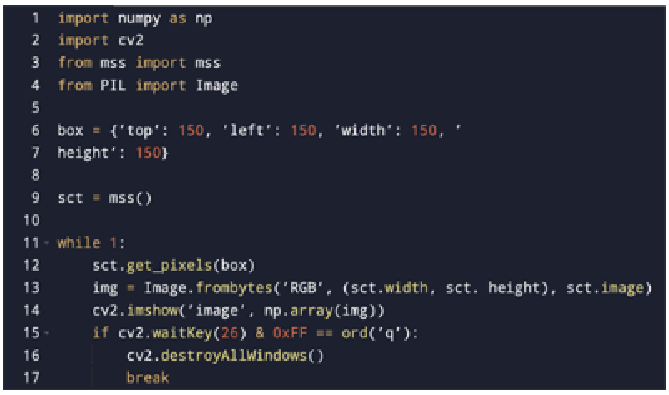
This paper explores the transformative power of classroom management software, examining its impact on teaching, learning, and student outcomes. We delve into critical features, benefits, and best practices, highlighting its potential to revolutionize education. The COVID-19 pandemic has accelerated online learning, revealing limitations in platforms like Zoom and Google Classroom [8]. Our team addresses this gap with an add-on app detecting faces, tracking attendance, and monitoring engagement in real time. Unlike existing apps, our tool ensures students stay focused and engaged during online classes.

While other researchers have made progress in visualizing online learning environments [5] and securing online classes [9], our focus is on real-time student attention and engagement detection, which is crucial for effective online learning.

METHODOLOGY

We have developed cutting-edge software that helps teachers monitor their students during online classes. Here's how it works: when a teacher starts an online session, our software starts recording a video of the students' screens. It then uses facial recognition to identify each student and track their engagement in real time. The software notes whether each student is paying attention, dozing off, or getting distracted and logs this information every few minutes. At the end of the class, the teacher receives a detailed report showing which students were engaged, who was sleeping, and when each student joined or left the class. In addition, our software has a real-time dashboard that uses green and red flags to indicate which students are on track and which ones need a nudge from the teacher. Its comprehensive feature sets our software apart from other virtual class monitoring tools. We have designed it to give teachers a complete picture of what is happening in their online classroom so they can take action to keep their students engaged and learning.

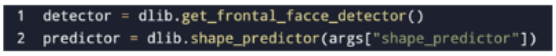
Screen Video. Our process involves capturing the teacher's laptop screen video during online classes, leveraging NumPy, a robust Python library for image processing. This enables us to manipulate and analyze the visual data, detecting and examining each student's face in the grid displayed on the teacher's screen. To enhance flexibility, we've integrated a feature allowing teachers to select a specific screen region for recording using a bounding box. This ensures that only the relevant area is captured, optimizing data collection for subsequent facial recognition and engagement analysis.



```
import numpy as np
import cv2
from mss import mss
from PIL import Image

box = {'top': 150, 'left': 150, 'width': 150, '
height': 150}

sct = mss()

while 1:
    sct.get_pixels(box)
    img = Image.frombytes('RGB', (sct.width, sct. height), sct.image)
    cv2.imshow('image', np.array(img))
    if cv2.waitKey(26) & 0xFF == ord('q'):
        cv2.destroyAllWindows()
        break
```

Figure 1 – Illustrates the code used for screen recording

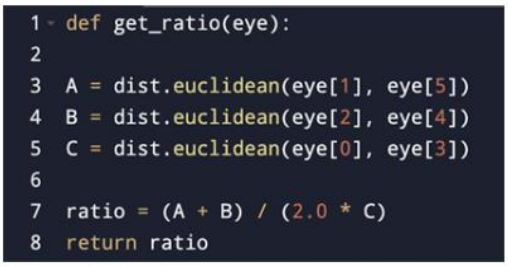
Face Detection. Our software technology records a live feed that extracts students' faces. Next, we identify each face using a state-of-the-art face identification method called Histogram of Oriented Gradients (HOG). Using this potent algorithm, we can determine which students actively participate in the online course and which do not. Using HOG, we can extract the most important visual information from the pictures, eliminating extraneous information and focusing on the key characteristics that make up a face. This method guarantees precise and effective face detection.



```
detector = dlib.get_frontal_facce_detector()
predictor = dlib.shape_predictor(args["shape_predictor"])
```

Figure 2 –Detection Module Code

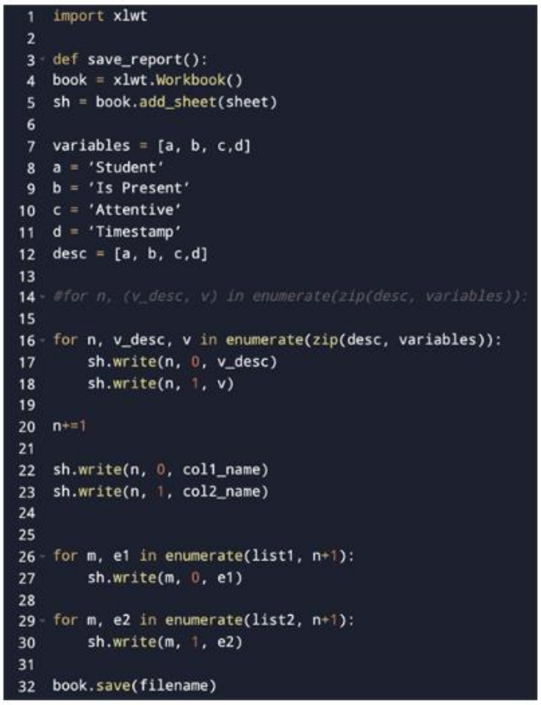
Drowsiness Detector. This is the primary component of our system, where we ascertain whether or not a student is involved. To begin, we determine the aspect ratio of the student's eye region by enlarging it. This indicates whether or not their eyes are open, and we utilize OpenCV to detect this. Additionally, we use SciPy to calculate the Euclidean Distance between facial landmarks, which aids in determining the direction of the student's gaze. Examining the ocular aspect ratio, we can decide if a learner is looking away, sleeping, or distracted. The student's face is then surrounded by a coloured box, with green denoting engagement and red denoting disengagement, using this understanding. Meanwhile, we keep a record of which students were attentive and which ones weren't. Figure 3 below shows the code that makes this drowsiness detection possible.



```
def get_ratio(eye):

A = dist.euclidean(eye[1], eye[5])
B = dist.euclidean(eye[2], eye[4])
C = dist.euclidean(eye[0], eye[3])

ratio = (A + B) / (2.0 * C)
return ratio
```

Figure 3 – Drowsiness Detection Code

Finalization. Following the video's acquisition, we examine all the data to obtain specific insights, such as who is present and attentive. We also show real-time results; we have to save this information into a file so the teacher can find a report on who was present and attentive after completing the online class. We save this report at a specific interval after analyzing the data.



```
import xlwt

def save_report():
book = xlwt.Workbook()
sh = book.add_sheet(sheet)

variables = [a, b, c,d]
a = 'Student'
b = 'Is Present'
c = 'Attentive'
d = 'Timestamp'
desc = [a, b, c,d]

#for n, (v_desc, v) in enumerate(zip(desc, variables)):

for n, v_desc, v in enumerate(zip(desc, variables)):
    sh.write(n, 0, v_desc)
    sh.write(n, 1, v)

n+=1

sh.write(n, 0, col1_name)
sh.write(n, 1, col2_name)


for m, e1 in enumerate(list1, n+1):
    sh.write(m, 0, e1)

for m, e2 in enumerate(list2, n+1):
    sh.write(m, 1, e2)

book.save(filename)
```

Figure 4 – Code for saving reports

RESULTS AND DISCUSSION

The design of the software is straightforward, and some coding knowledge is needed to start it. Teachers need to run the start code to start the system. It will then work automatically, and after closing the software, it will save the result on the desktop. The computer where this software will be used needs some pre-installed software. As we are using Python language to develop this software, the user of this software must install Python on his/her computer to run this software. The software works very simply. After opening the software, it will ask the user if the person is ready to start. Once the user turns it on, the video call will automatically show green and red boxes.

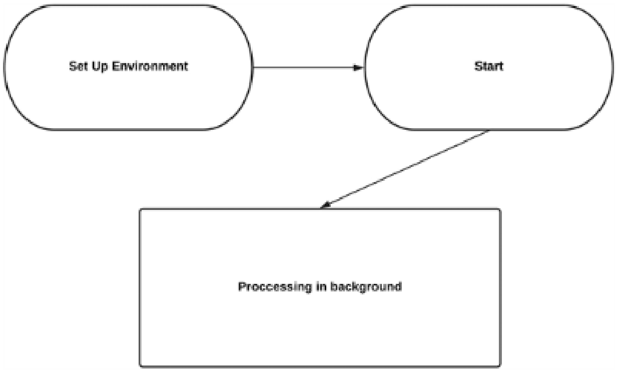


Figure 5 – Controller Design and Implementation

Python is utilized in the software's construction, and numerous libraries are employed to ensure flawlessness. The method operates in stages. Taking a video off the screen is the initial stage; processing and picture analysis are the next steps. There are two sorts of processing in the process section: face detection, which tells us who is there and who is not, and sleepiness detection. Following the analysis, we present the outcome in two formats. In the first, boxes are displayed on the screen, while in the second, the result is saved in a file for the teacher to view after the lesson.

The following figure shows a flowchart showing the overall steps to implement the application. Figure 6 shows the flowchart of the controller design.

Testing. The paradigm was displayed in a few college projects that displayed presentations before the application's execution. Both the college staff and the visitors gave us very positive reviews. After usage, this application's clearance was finalized by continually allowing a few beta clients to find new problems and errors. Following the acceptance of a few surveys from the beta clients, the application was debuted at multiple public and international events.

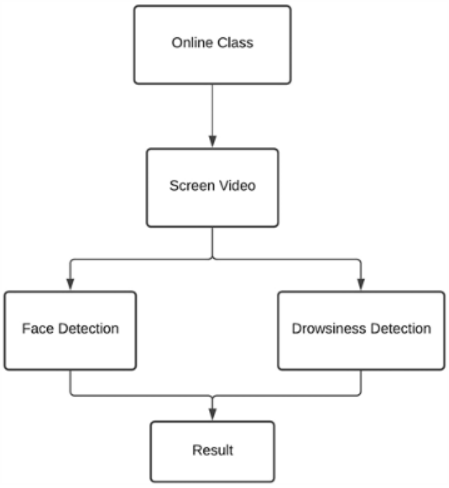


Figure 6 – Flowchart of controller design

Testing is done very quickly. After launching our software, we initiate a random video conference with at least three participants. We verify whether or not the application is detecting faces in the following testing phase. Our first test case is successful if it accurately detects every face. It will be our second test case if it can determine whether or not a person's eye is closed. We go on to our third test scenario if it passes this one test case, which detects drowsiness. By these simple test cases, we can test this software.

Feasibility and Cost Analysis. The computer runs at a steady pace with exceptional smoothness. As time went on, improvements in internal infrastructure were made to ensure that users had an excellent hands-on experience, as well as increases in the performance speed of all varieties of computers.

Memory Management. The board, or memory, is crucial in ensuring that any application runs well. The internal memory, which plays a vital role in handling the cycles, administrations, and apps introduced by the client or currently existing on the device, is thus the critical differentiator between this application and other accessible applications. We must have a lot of memory for this software; for instance, taking screen videos requires a lot of memory space in addition to analysis. Following analysis, the real-time outcome must be displayed. Nonetheless, the application is designed to manage every challenge and provide our cherished teachers with a highly user-friendly experience.

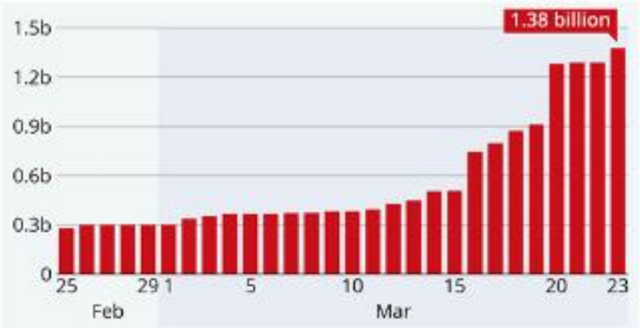
Database. Only simple databases, such as SQLite, more frequently utilized in embedded system applications, are employed. This is significant because of its data transaction structure, which makes data management on the phone and in the application more effective.

Version. This works perfectly in all computer versions with little or no lags. It is also supported in Mac OS and Linux, supporting all computers. However, the laptop should have Python installed to run it. This application was developed using Python 3.5. Also, to run the software on a computer, you must install Python 3.5. Without this, it will not work. It will run very smoothly with the correct Python 3.5.

Cost. This program can be used without additional costs; anybody can use it. Additionally, they can alter it and grow further to their needs. Additionally, since the teacher is now engaged in a video conference, he already has an active internet connection; the application will function properly even if the user does not.

Thus far, its easy-to-use features and low maintenance costs allow us to regulate the application's viability and cost analysis confidently.

Use of the Application. Although a drowsiness detector is mainly used for the driver's safety in the vehicle, the primary purpose of this program is to assess a student's attentiveness in an online classroom, which has considerably grown due to COVID-19. According to UNESCO and a Statista chart, 1.38 billion students were influenced by COVID-19 between February 25 and March 23. This is seen in Figure 7 below.



Figures refer to learners enrolled at pre-primary, primary, lower-secondary, and upper-secondary levels of education, as well as at tertiary education levels.
Source: UNESCO

Figure 7 – Number of learners impacted by national school closures worldwide

CONCLUSIONS

According to this study, software can guarantee that every student in an online course is paying attention. This software will automatically record attendance, which will benefit them. This essay addresses the program's functionality, namely how it can identify which pupils are paying attention and which are leaving early from class. The paper thoroughly explains how the software will adequately handle attentive face detection and attendance. In the paper, we examine the idea of developing a mobile app so that educators who use mobile apps for online classes can utilize the program, as well as the possibility of creating a helpful user interface (UI) so that educators can use it more naturally.

REFERENCES

1. Marzano, R. J. (2003). *Classroom management that works: research-based strategies for every teacher*. Retrieved from https://perino.pbworks.com/f/4%20CLM-Basic-Text(M).pdf

2. Hattie, J. (2012). *Visible Learning for Teachers: Maximizing Impact on Learning*. Routledge.

3. Wang, M. C., & Holcombe, R. (2010). Classroom Management and Student Achievement. *Journal of Educational Psychology, 102*(2), 281-291.

4. Emmer, E. T., & Stough, L. M. (2001). Classroom Management: A Critical Part of Educational Psychology, With Implications for Teacher Education. *Educational Psychologist, 36*(2), 103–112. doi: 10.1207/s15326985ep3602_5

5. Kapoor, A., & Picard, R. W. (2005). Multimodal affect recognition in learning environments. *Proceedings of the 13th Annual ACM International Conference on Multimedia*. doi: 10.1145/1101149.1101300

6. Siemens, G., & Baker, R. S. J. d. (2012). Learning analytics and educational data mining. *Proceedings of the 2nd International Conference on Learning Analytics and Knowledge*. doi: 10.1145/2330601.2330661

7. Whitehill, J., Serpell, Z., Lin, Y.-C., Foster, A., & Movellan, J. R. (2014). The Faces of Engagement: Automatic Recognition of Student Engagement from Facial Expressions. *IEEE Transactions on Affective Computing, 5*(1), 86–98. doi: 10.1109/taffc.2014.2316163

8. Yang, X., Cheng, P.-Y., Lin, L., Huang, Y. M., & Ren, Y. (2018). Can an Integrated System of Electroencephalography and Virtual Reality Further the Understanding of Relationships Between Attention, Meditation, Flow State, and Creativity? *Journal of Educational Computing Research, 57*(4), 846–876. doi: 10.1177/0735633118770800

9. Pham, H. C., Brennan, L., & Furnell, S. (2019). Information security burnout: Identification of sources and mitigating factors from security demands and resources. *Journal of Information Security and Applications, 46*, 96–107. doi: 10.1016/j.jisa.2019.03.012