Scientific Research Publishing

Journal of Software Engineering and Applications, 2021, 14, 83-94
https://www.scirp.org/journal/jsea
ISSN Online: 1945-3124
ISSN Print: 1945-3116

Development and Analysis of a Machine Learning Based Software for Assisting Online Classes during COVID-19

Tasfiqul Ghani, Nusrat Jahan, Mohammad Monirujjaman Khan, S. M. Tahsinur Rahman, Sabik Tawsif Anjum Islam

Department of Electrical and Computer Engineering, North South University, Bashundhara, Dhaka, Bangladesh
Email: monirujjaman.khan@northsouth.edu, smtahsinur@gmail.com, sabik.tawsif@northsouth.edu

How to cite this paper: Ghani, T., Jahan, N., Khan, M.M. Rahman, S.M.T. and Islam, S.T.A. (2021) Development and Analysis of a Machine Learning Based Software for Assisting Online Classes during COVID-19. *Journal of Software Engineering and Applications*, **14**, 83-94.
https://doi.org/10.4236/jsea.2021.143006

Received: March 1, 2021
Accepted: March 27, 2021
Published: March 30, 2021

Abstract

Amid the Covid-19 widespread, it has been challenging for educational institutions to conduct online classes, facing multiples challenges. This paper provides an insight into different approaches in facing those challenges which includes conducting a fair online class for students. It is tough for an instructor to keep track of their students at the same time because it is difficult to screen if any of the understudies within the class are not present, mindful, or drowsing. This paper discusses a possible solution, something new that can offer support to instructors seeing things from a more significant point of view. The solution is a facial analysis computer program that can let instructors know which students are attentive and who is not. There's a green and red square box for face detection, for which Instructors can watch by seeing a green box on those mindful students conjointly, a red box on those who are not mindful at all. This paper finds that the program can automatically give attendance by analyzing data from face detection. It has other features for which the teacher can also know if any student leaves the class early. In this paper, model design, performance analysis, and online class assistant aspects of the program have been discussed.

Keywords

Online Class, Python, Technology, Artificial Intelligence, Analysis, Machine Learning, Covid-19, Software, Face Detection, Drowsiness Detector

1. Introduction

Due to this coronavirus circumstance, nearly every nation has confronted numerous changes. The lockdown has made everyone's life so challenging that regular individuals faced standard of living challenges. Students are taking classes remotely, and teachers are attempting to discover the lasting strategy of making the classes smooth and viable. This paper finds that the program in question diminished the chances of making excuses. As a result, the students started taking classes seriously. This program allows any teacher to take classes smoothly.

Previously, different programs have been developed for online class monitoring but this paper discusses about introducing a new technology which is based on machine learning with face detection. This detection involves observing a face, detecting eye position, and observing eye blinking patterns. The whole process is done by using python. To detect fatigue, a webcam has been used towards students' faces, and only teachers can recognize that seeing the boxes on the face means not attentive, students might fall asleep, and the green box on the face means students are attentive and doing their classes. We come across only these two colors because a teacher has to see all the students and one frame, and each student gets a minimal space within this frame. So, it would be easier for teachers to observe if the eyes are closed for a particular period, then a red box pops up on that particular student's face. The development of technologies for detecting or preventing drowsiness at the wheel is a very tough challenge. Keeping students engaged and motivated is not an easy task to do. Student distraction occurs when an object or event draws a person's attention away from the online classroom. This software will bring back and gives both teachers and students a real-time face to face feeling classroom.

Given the current Covid-19 scenario, everyday life is solely dependent on the virtual world and a virtual classroom is no exception. During this time of pandemic, it has come to our understanding that adopting to a new normalcy is the ideal solution for surviving. With that online class became the only option for education. In order to maintain the proper quality of a classroom, a proper management software is required and this software gives the edge both for the teachers and the students.

Between 8th of March and 6th of December 2020, according to the DGHS Press Release there were four hundred seventy-seven thousand five hundred forty-five (477,545) COVID-19 confirmed by rRT-PCR, including six thousand six hundred forty-four (6838) related deaths (CFR 1.43%). Bangladesh is the top 26th country in the world and accounts for 0.75% of the COVID-19 disease burden in the world. **Figure 1** below is showing the trend of COVID-19 cases and deaths, 08 March-06 December 2020, Bangladesh. [1]

In this paper, we will discuss and show the possible ways where taking an online class would be more efficient for an instructor ensuring all the protocols are followed properly. This paper will solely focus on the development of a computer program which can ensure that the students are attentive in class, in addition to taking automatic attendance and students remain in class without the use of any unfair means.

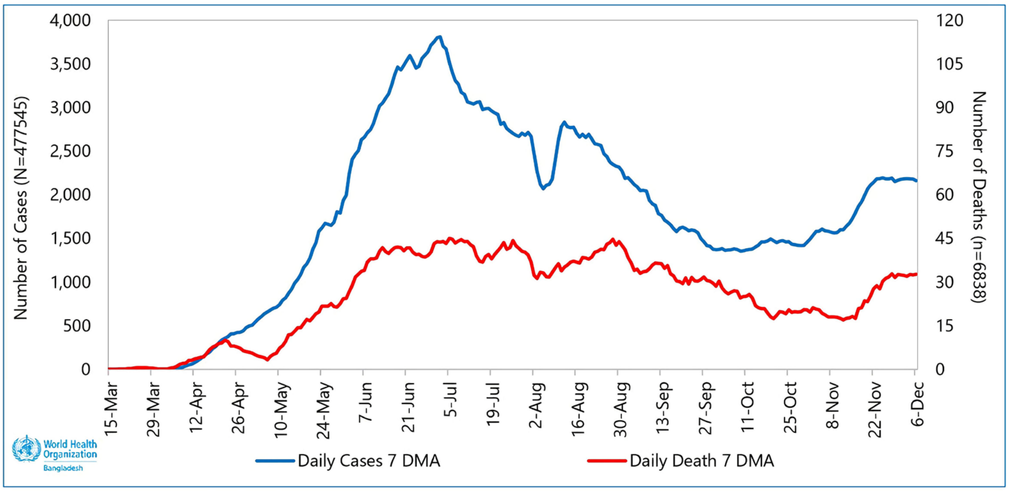


Figure 1. Covid-19 trend cases and deaths of Bangladesh [1].

2. Related Works

We cannot deny the fact. Suddenly, because something is totally out of control to the global and students are learning online, educators and administrators are continuously searching for permanent solutions of new tools, technologies, and solutions to face challenges that appeared with COVID-19 they have not faced before. According to Zoom, the most used video calling platform, Zoom had 12.92 million monthly active users in February 2020 [2]. Moreover, in these calls, above 30% is used for online classes. There are many tools or platforms for video calling, but none have a feature to detect drowsiness and student attention. There are also some tools like Google classroom [3], which has many features for class management, but they also do not have any feature to detect students and give attendance or detect which students are attentive. Previously work visualization has been done [4], which can be used by the teachers to observe students. Various domains have been used in visualization techniques to interpret happenings during an online class.

In Google, zoom, or other traditional video calling app no feature can detect whether the users in the call are attentive or not, or the users are feeling Drowsiness. These apps are only used for creating meetings or single calls that's it. The main usage of our application is, it's an add-on app for traditional video calling applications. Our application will work parallel with these. Our application will be running when the user in a video call using google meets or zoom and our app will detect the faces also other things will be done. The main difference between google meet and our application is, Google meet can not detect face and tell that these are the persons who are available in the call, Also Google meets can not detect whether the users are attentive or sleeping. The same thing goes for other traditional applications. Those applications are very good and scaleable for video calling but these features are not available. Our main focus of the research was how we can detect users and their state in realtime which is not available in the traditional calling applications.

Framework based works have been done too previously to monitor students in online classes, which is secure against unauthorized logins [5]. The technique used was Liveness Face Detection technique and as LFD [6] has been used so IPS has no effect on duplicate. Although more work was left on the synchronous learning environment.

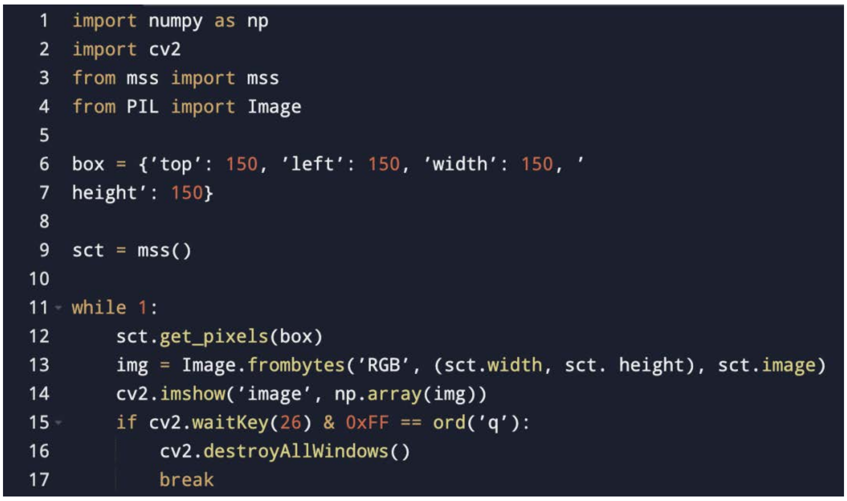
3. Methodology

A list of specific methodologies has been applied to make this software. When a teacher is in an online class, the software is active it takes screen video of the computer so in screen video software found many students faces .then the software first check every student's face and detect their ID and make a note with time in the same time it keep note the status of the student whether the student is attentive or sleeping or not looking at the webcam or his screen. Every 5 or 10 minutes, the software analyzes it and keeps the notes. After class ends, the software shows a report to the teacher that students were attentive who were sleeping, and each student's entry and exit time. The software also shows a real-time box with green and red color, indicating which students are attentive, which are not so the teacher can warn them instantly during the class.

There are many other Virtual Class Monitoring software available on Internet these days which are limited to a few services

3.1. Screen Video

The first step is to take the screen video of the teacher's laptop. As the teacher is in an online conference so on the teacher's screen, it is showing a list of student's faces. So, when the software takes screen video, it finds all the faces which will be analyzed. For this, we have used NumPy [7], which is a python library to do lots of image processing. We can also set a bounding box during the screen record to allow the teacher to take only a specific area. Following **Figure 2** shows the code for screen recording.



```
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 2. Screen recording code.

3.2. Face Detection

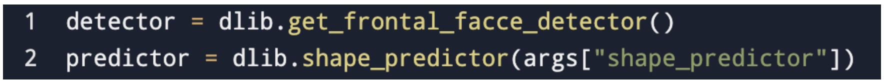
The first step is to take the video and get the students' faces. The second part of the system is to detect those faces. To detect faces, we have used Histogram of Oriented Gradients [8] based face detection. Using this formula, we detect which students are currently present in the online class and which students are not present. Using a Histogram of oriented gradients [9], we fetch the vital information of the images and avoid fewer essential things which are with this image, which is very efficient to detect a face. Following **Figure 3** shows the code for face detection.

3.3. Drowsiness Detector

This is the most crucial part of the system. In this part, we detect if the student is sleeping or looking outside, or not attentive. First, we detect the student's eye region, and then we find out the aspect ratio of the eye region. Using the eye region's aspect ratio, it is very easy to determine if the eye is closed or open. We have used OpenCV [10] to detect the eye region and the eye's aspect ratio in this part [11]. We will also use SciPy [12]. Using SciPy [13], we will find out the Euclidean Distance [14] of the facial landmark. It is beneficial to find out the aspect ratio of the eye. Using the student's eye aspect ratio, we can find out where he/she is looking at. Also, we can detect whether he is sleeping or not and more information. Using this information, we will show a box to the student's face. If the box is green, the teacher will understand he is attentive; if the box is red, the teacher can detect instantly that the student is not attentive, and then the teacher can warn him. In the meantime, we will note which students were attentive, which students were not attentive. Following **Figure 4** shows the code for drowsiness detection.

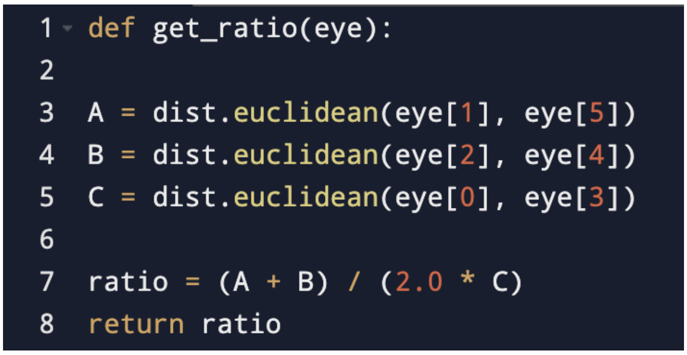
3.4. Finalization

After getting the video, we analyze all the data and get some information like



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

Figure 3. Face detection code.



```
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
```

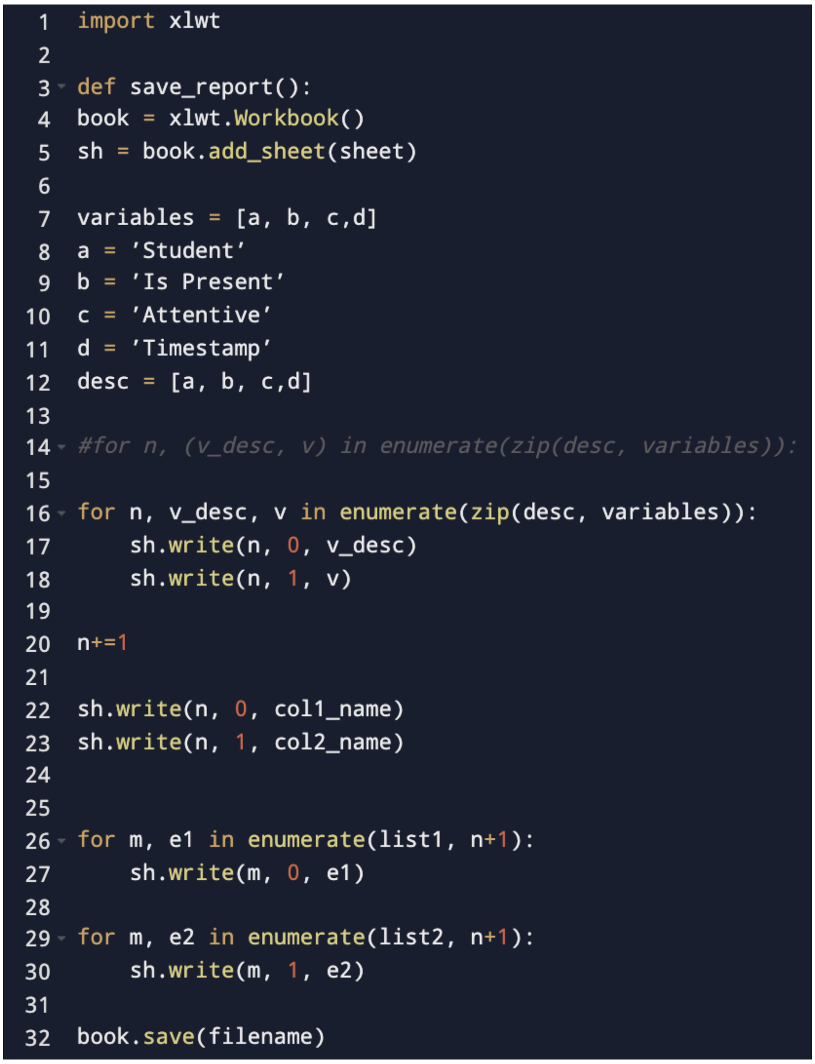
Figure 4. Drowsiness detection code.

who is present and attentive. We are also showing real-time results; also, we have to do another thing, which is saving this information into a file so that after completing the online class, the teacher can find a report on who was present and attentive. We are saving this report in a specific interval after analyzing the data. Following Figure 5 shows the code for saving reports.

4. Model Design

The design of the software is very simple and needs some coding knowledge 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 in his/her computer to run this software (Figure 6).

The software works in a very simple way. After opening the software it will



```python
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 5. Saving report code.

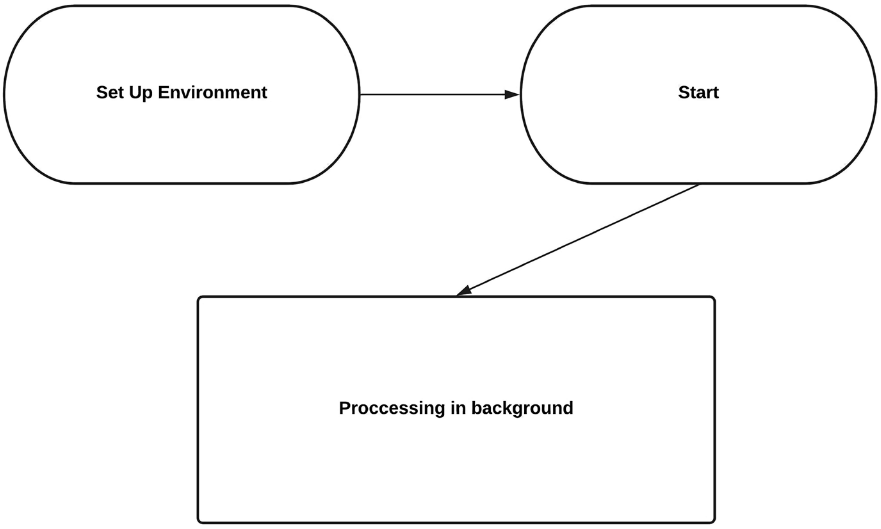


Figure 6. Model design.

first ask the user if the person is ready to start or not. Once the user turns it on, it will automatically start showing green and red boxes in the video call.

5. Controller Design and Implementation

The software is built using python; we have also used many libraries to make it perfect and smooth. The system works in some steps. The first step is taking video from the screen; the second step is to analyze the images from the video and process it. In the process section, there are two types of processing: first one is face detection [15], which lets us know who is present and who is not, and the second one is drowsiness detection. After the analysis, we show the result in two ways. The first one is showing boxes on the screen, and the second one is saving the result in a file so that the teacher can see it at the end of the class. The following figure shows a flowchart to show the overall steps followed to implement the application. Following Figure 7 shows the flowchart of controller design.

6. Testing

Preceding the application's execution, the model was shown in a couple between college ventures displaying presentations. We got much good feedback from the guests and the college employees.

Later upon usage, this application's approval was completed by permitting some beta clients to discover new issues and mistakes consistently. In the wake of accepting a few surveys from the beta clients, the application was introduced in numerous public and global shows.

The process of testing is quite simple. First, we start our software, and then we start a random video call of at-least 3 people. In the next step of testing, we check whether the application is detecting faces or not. If it detects all the faces correctly then our first test case is passed. Our second test case is whether it can detect a person's eye is closed or not. If it passes this test case we go for our third

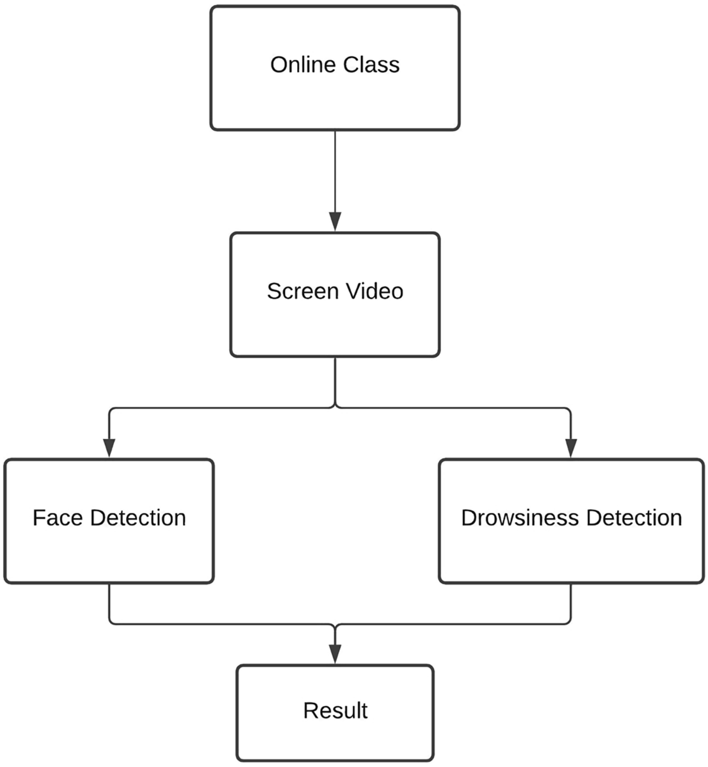


Figure 7. Flowchart of overall controller design.

test case which is detecting drowsiness. By these simple tests cases we can test this software.

7. Feasibility and Cost Analysis

Computer performs very smoothly at a constant speed. As days passed, the developments of all types of computers were not just limited to its performing speed, but there were also advancements in internal infrastructure to make sure that people have a great hands-on experience.

7.1. Memory Management

Memory, the board, assumes an imperative part to guarantee the smooth presentation of any application. Accordingly, the significant distinction between this application and other accessible applications is the internal memory that assumes a significant part in dealing with the cycles, administrations, and applications introduced by the client or are now present in the gadget. In this software, we need to be memory-hungry; for example, taking screen video is a vast work for memory also, we need to analyze it. After analyzing we need to show the real-time result. However, the application is built to handle all the obstacles and make it very user friendly for our beloved teachers.

7.2. Database

We are only using basic databases like SQLite, which are more widely used in embedded system applications [16]. This is important because of its data transactions structure, which allows the data to be more efficiently managed within the phone and the application [17].

7.3. Different Platform Versions

This works perfectly in all the computer versions with little or no lags at all. It is also supported in Mac OS and Linux. So, it supports all the computers. However, the computer should have python installed to run it. Following **Figure 8** shows the users of different OS.

The development of this application was carried out in Python 3.5. Also, to run the software on a computer is required to install Python 3.5. Without this, it will not work. It will run very smoothly with the correct python 3.5.

7.4. Cost

There is no additional cost to run this application; anyone can use it without any cost. Also, they are allowed to modify it and develop more according to their need. Also, as the teacher is already in a video call, he already has an active internet connection; even if the user does not have any internet connection, it will work fine.

So far, we can say that the application's feasibility and cost analysis can be regulated at ease due to its user-friendly features and no cost of maintenance. **Figure 9** shows a normal online class scenario.

8. Use of the Application

Although a drowsiness detector is mostly used for the safety of the driver in the

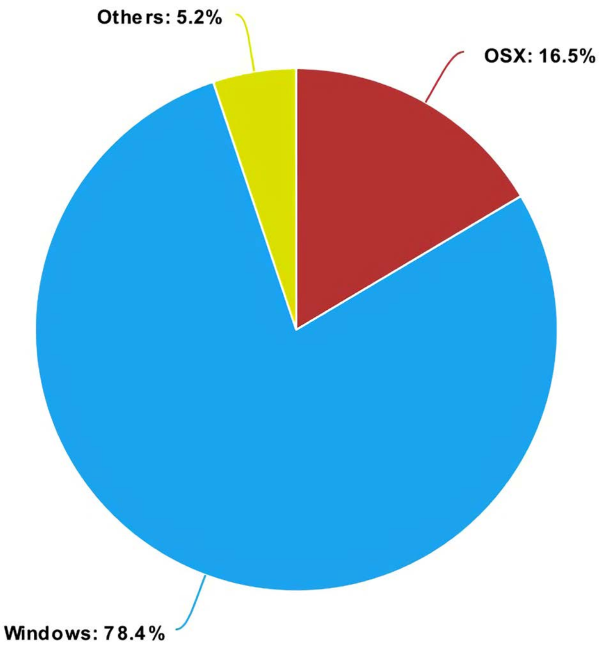
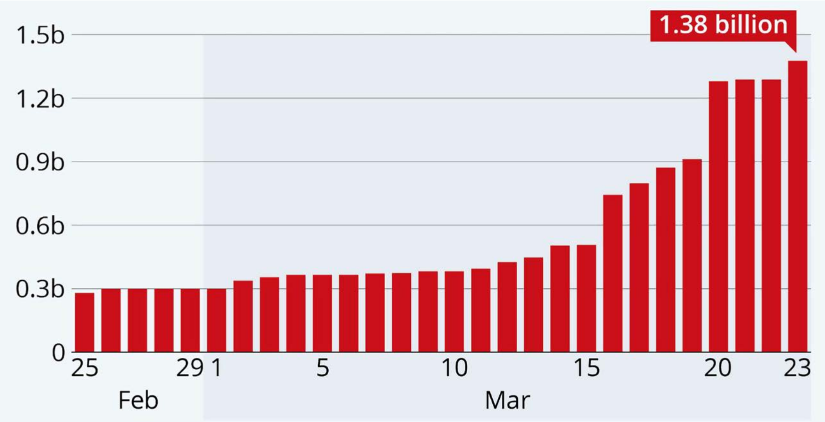


Figure 8. Users of different OS.

Figure 9. A normal online class scenario.



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 10. Number of learners impacted by national school closures worldwide [19].

car. This application is mainly made for the online class system to check the attentiveness of the student. Because of Covid-19 has increased dramatically. According to UNESCO and a chart made by Statista it is seen that from February 25 to March 23 the Covid-19 has impacted 1.38 billion learners. In the below figure it can be seen in **Figure 10** [18]. And In this situation, online classes are the only solution. And this application will help make the online classes more helpful to the teacher and will create a more effective online class environment. The following figure shows the statistics of impact of the school closures worldwide [19].

9. Conclusion

This paper finds that the software can make sure that all the students are attentive in an online class. This software will take attendance automatically so that it will be constructive for them [20]. This paper discusses the functional ability of the program where it will detect which students are attentive or not; it will also detect which students are leaving before the class ends. The paper discusses in details where the software will take care of the attendance, attentive face detection successfully. In the paper, we discuss the possibility of adding a useful UI so that teachers can use the program more intuitively, in addition to the development of a mobile so that teachers who use mobile apps for online classes can use it. This paper ensures that that bunking online classes will be reduced and students will be more active throughout the class and respond that way.

Acknowledgements

The authors would like to thank Department of Electrical and Computer Engineering at North South University of Bangladesh.

Conflicts of Interest

The authors declare no conflicts of interest regarding the publication of this paper.

References

[1] World Health Organization. https://www.who.int/bangladesh/emergencies/coronavirus-disease-(covid-19)-update/coronavirus-disease-(covid-2019)-bangladesh-situation-reports

[2] Zoom User Statistics. https://www.cnbc.com/2020/02/26/zoom-has-added-more-users-so-far-this-year-than-in-2019-bernstein.html

[3] Google Classroom. https://support.google.com/edu/classroom/answer/6020279?hl=en

[4] France, L., Heraud, J.-M., Marty, J.-C., Carron, T. and Heili, J. (2006) Monitoring Virtual Classroom: Visualization Techniques to Observe Student Activities in an e-Learning System. *6th IEEE International Conference on Advanced Learning Technologies* (*ICALT'*06), Kerkrade, 5-7 July 2006, 716-720. https://doi.org/10.1109/ICALT.2006.1652543

[5] Satapathy, A. and Livingston, J. (2018) An Intelligent Framework Prototype for Monitoring Students in Virtual Classroom. *Indonesian Journal of Electrical Engineering and Computer Science*, **12**, 1151-1158. https://doi.org/10.11591/ijeecs.v12.i3.pp1151-1158

[6] Hamdi, *et al.* (2020) Liveness Detection with OpenCV. PyImage Search. http://www.pyimagesearch.com/2019/03/11/liveness-detection-with-opencv

[7] Numpy. https://numpy.org/

[8] Histogram of Oriented Gradients. https://www.pyimagesearch.com/2014/11/10/histogram-oriented-gradients-object-detection/

[9] Yamasaki, T. (2010) Histogram of Oriented Gradients (HoG). *Journal of the Institute of Image Information and Television Engineers*, **64**, 322-329. https://doi.org/10.3169/itej.64.322

[10] OpenCV. https://opencv.org

[11] Khan, M., Chakraborty, S., Astya, R. and Khepra, S. (2019) Face Detection and Recog-

nition Using OpenCV. 2019 *International Conference on Computing, Communication, and Intelligent Systems* (*ICCCIS*), Greater Noida, 18-19 October 2019, 116-119. https://doi.org/10.1109/ICCCIS48478.2019.8974493

[12] SciPy. https://www.scipy.org/

[13] Al Omran, F.N.A. and Treude, C. (2017) Choosing an NLP Library for Analyzing Software Documentation: A Systematic Literature Review and a Series of Experiments. 2017 *IEEE*/*ACM* 14*th International Conference on Mining Software Repositories* (*MSR*), Buenos Aires, 20-21 May 2017, 187-197. https://doi.org/10.1109/MSR.2017.42

[14] Glossary. Euclidean Distance (n.d.). http://rosalind.info/glossary/euclidean-distance/

[15] Sharif, M., Ayub, K., Sattar, D. and Raza, M. (2012) Real Time Face Detection. *Sindh University Research Journal* (*Science Series*), **44**, 597-600.

[16] Chen, D., Han, X. and Wang, W. (2010) Use of SQLite on Embedded System. 2010 *International Conference on Intelligent Computing and Cognitive Informatics*, Kuala Lumpur, 22-23 June 2010, 210-213. https://doi.org/10.1109/ICICCI.2010.79

[17] Siebes, A. and Kersten, R. (2011) A Structure Function for Transaction Data. *Proceedings of the* 11*th SIAM International Conference on Data Mining*, 558-569. https://doi.org/10.1137/1.9781611972818.48

[18] Li, C. and Lalani, F. (n.d.) The COVID-19 Pandemic Has Changed Education Forever. This Is How. https://www.weforum.org/agenda/2020/04/coronavirus-education-global-covid19-online-digital-learning/

[19] McCarthy, N. and Richter, F. (2020) Infographic: COVID-19's Staggering Impact on Global Education. https://www.statista.com/chart/21224/learners-impacted-by-national-school-closures/

[20] Ofualagba, G., Osas, O., Orobor, I.A., Oseikhuemen, I. and Etse, O. (2018) Automated Student Attendance Management System Using Face Recognition. *International Journal of Educational Research and Information Science*, **5**, 31-37.