ISSN 2278-3091
Volume 9, No.4, July – August 2020
International Journal of Advanced Trends in Computer Science and Engineering
Available Online at http://www.warse.org/IJATCSE/static/pdf/file/ijatcse191942020.pdf
https://doi.org/10.30534/ijatcse/2020/191942020

Class-EyeTention: A Machine Vision Inference Approach of Student Attentiveness' Detection

Jennalyn N. Mindoro[1], Moises Fernandez Jardiniano[2], Nino U. Pilueta[3], Honeylet D. Grimaldo[4], Dennis P. Ordovez[5]
[1]Technological Institute of the Philippines, Manila Philippines, jnicolas.cpe@tip.edu.ph
[2] FEU Institute of Technology, Manila Philippines, moises.jardiniano@gmail.com
[3] FEU Institute of Technology, Manila Philippines, nupilueta@gmail.com
[4] FEU Alabang, Alabang Muntinlupa, Philippines, grimaldohoneylet@gmail.com
[5] EARIST Manila, Manila Philippines, dennisordovez@gmail.com

ABSTRACT

Active learning is the key to effective learning in the classroom. Analyzing student participation is very critical in enhancing learning as well as teaching. Attention is the key component in successful learning. However, it is difficult to monitor the attention of individual students in the classroom by using self-reporting. Furthermore, the use of traditional CCTV cameras or video surveillance is intended to reduce the effort involved in manual checking as effectively as possible. In consideration of these difficulties, the use of machine vision to detect student knowledge during the class discussion was suggested. In this research, facial recognition is enabled by an image that has been obtained from student images. The study used three (3) methods to monitor student attention: image detection, webcam detection, and video detection. The aim of this is to continue to process the video captured using traditional CCTV cameras. This will also produce a report on student behavior: attention and not attention. The generated report of the student behavior was based on attention and non-attention level. TensorFlow Lite and Raspberry PI were implemented in this study. In the generated report, students will be informed of one's participation in the class discussion utilizing a time stamp providing details on the behavioral activity conducted by the student. Results show that face recognition results in an accuracy of 96.7% and the detection of attentiveness were also performed.

Key words: Image Detection, Face Recognition, Raspberry Pi, Student Attentiveness, TensorFlow Lite.

1. INTRODUCTION

The ability to predict student performance has been the subject of growing concern. Knowledge of anticipated academic success is also useful feedback for educators and school administrators. This knowledge may be used to identify and target vulnerable students at risk of dropping out or in need of extra care [1]. The cognitive and motivational consistency of classroom learning is of vital importance to students' emotions. Positive emotions impact learning by influencing student understanding, motivation, use of learning methods and self-regulation of learning [2]. Attention is the key component in active learning. It is better defined as the continued concentration of cognitive resources on knowledge while avoiding distractions. Attention or diligence is used to characterize the ability to sustain focus during long periods of time, such as during classroom lectures. However, the process of monitoring the attention of individual students in the classroom through self-reporting is difficult. In addition, this interferes with the learning process, which is also the case with the use of psychophysical data sensors [3]-[4].

Teachers must be able to observe the student's behavior and understand the appropriate cues for the student's actual involvement in learning activities. Teacher training especially the inexperienced teachers can allow to develop these skills by using videotaped teaching to highlight which indicators should be considered. However, this presupposes that reliable measures of student interest in learning are established, and video work is planned as efficiently as possible to minimize the effort involved in manual coding and video-examination [5]. The use of traditional methods is still difficult to track and predict student actions, particularly in most circumstances where there are many students or in broad areas. One way to address these challenges is to use the technological advancements made in recent years in fields such as computer vision to implement other analytical techniques.

Recent advances in visual sensors and computer vision approaches have made it possible to track automatic behavior and the emotional state of learners at various levels. In addition, the use of machine learning algorithms and deep learning are the approaches used in monitoring systems to produce excellent results [3]. Many sophisticated algorithms can be used on the Raspberry Pi for various applications in many areas [6] which can be used to track and include the involvement of assessment students in each scenario. The use of Face Recognition (FR) and Face Detection are fields that have become more relevant in the field of Computer Vision (CV) since ancient times [7]. Having these allows integrating

such methods in developing a system that will monitor student behavior, particularly is the student is attentive or not.

In this study, face recognition is triggered by an image that has been collected from student images. The generated data were annotated, divided into training and testing datasets using TensorFlow. The developed model has been implemented in Raspberry Pi. The integrated-in camera was used to capture student images while the class was going on. The study employed three (3) methods to track student attention: image detection, webcam detection, and video detection. The aim of this is to continue to process the video captured using traditional CCTV cameras. This will also produce a report on student behavior: attention and not attention. This will still generate a report of the student behavior: attention and not attention. Using the generated report, students will be informed of one's participation in the class discussion using a timestamp providing details on the behavioral activity conducted by the student. The results show that this study shows successful monitoring of the response in real-time with increased recognition levels. In summary, the authors proposed a methodology of that integration of the TensorFlow lite algorithm and raspberry pi in determining the student behavior (attention and non-attention level) inside the classroom based on facial behavioral cues. The study integrates a face-recognition and behavioral understanding that can efficiently monitor classroom events and conduct student evaluations.

2. REVIEW IN RELATED LITERATURE

The first conventional semi-automated FR system was developed in 19601 to locate the characteristics of a person. FR is a biometric framework that recognizes an individual face in a digital image by analyzing and comparing patterns. There are typically three stages in the Face Recognition System (FRS); Acquisition is the identification and capture of facial descriptions from different angles; Normalization is the segmentation, organization, and accuracy of facial descriptions; Identification is an example, a rendering of unknown facial descriptions and links them to well-known identity models [8]. Facial recognition focuses on the origin of human facial expression. Spotlight extraction techniques identify and focus on a single type of facial spotlight. The facial recognition system uses machine-based learning techniques to arrange highlights grouped together into one of the groups (with an individual) in the database. Face Identification is one of the most commonly used forms of biometry[9].

Several studies have been applied to these techniques. For security applications, home and building security against a human intruder is a common application of face recognition using PI[10]. The proper illustrative approach has been implemented in such a way that the intrusion detection is effective, and the images thus obtained are transparent enough that the intruder is visible. The device is designed as a smart mirror that provides both knowledge and home security. The device is designed to accept touch and smartphone commands and the accuracy was measured up to 96% [11]. As part of IoT, motion detection and face recognition using Raspberry Pi has been used in studies. These are defined in the sense of the creation of a network of intelligent solutions for home use that can be used as a single stand-alone functional unit or as an element of a larger system linked to the Internet of Things [12].

Several studies have also shown that raspberry pi can be greatly used in other areas. A mobile robot platform with Arduino Uno and raspberry pi has been developed for autonomous navigation. The mobile robot will travel into 2D environments as a line tracker robot with tracking, navigation and obstacle-avoidance features [13]. A web-based application model to monitor bridge travelers using Raspberry Pi has been developed for web-based applications. This is a web-based framework for automating travelers in all possible movements of its components[14]. A new data logger based on Raspberry-Pi to track the locomotion of Arctic invertebrates has been developed, tested, and deployed in the field for data analysis. The system uses infrared sensors to check-in Vivo invertebrates pick up the behavior of the locomotive, data is collected for analysis [15]. Raspberry Pi has also recently been used in chromatographic analysis. A low-cost, open-source wireless strip chart recorder for chromatographic detectors using Raspberry Pi was studied. A complete user interface to control the device has been created to enable the collection, filtering and processing of chromatographic data [16]. For the attendance system, an implementation of the business architecture of a low-cost Facebook status monitoring system is used as an individual attendance status via social media status messages. The Facebook Graph API was used to view app status notifications to show on the LCD screen [17]. It was also used in early detection of using raspberry PI, image processing and CNN [18].

In the area of attentiveness detection, the student is either inattentive to fatigue or to distraction. The numerous attention patterns involved in the classroom are warning, exhausted, sleepy, busy, leaving the classroom, disappearing from the view of the camera, reacting to sessions, and catch-up [19]. Moreover, this was used in attendance monitoring using image processing and YOLOv3[20]. Another important application of the monitoring behavior was in dam construction sites. The feasibility of a real-time monitoring system provides prompt analysis support of workers' behavior via ZigBee-based tracking technology [21] and also used for object detection [22].

3. METHODOLOGY

3.1 Conceptual Framework

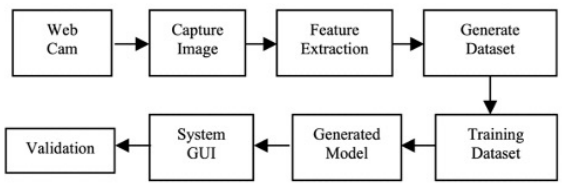

Figure 1: Student Behavior Monitoring Conceptual Framework

Figure 1 illustrates the conceptual framework of the study. Several images were captured during the three (3) hour class discussion using a webcam. Students were instructed to act naturally while the collection of data is being done. Figure 2 below shows the proposed method used.

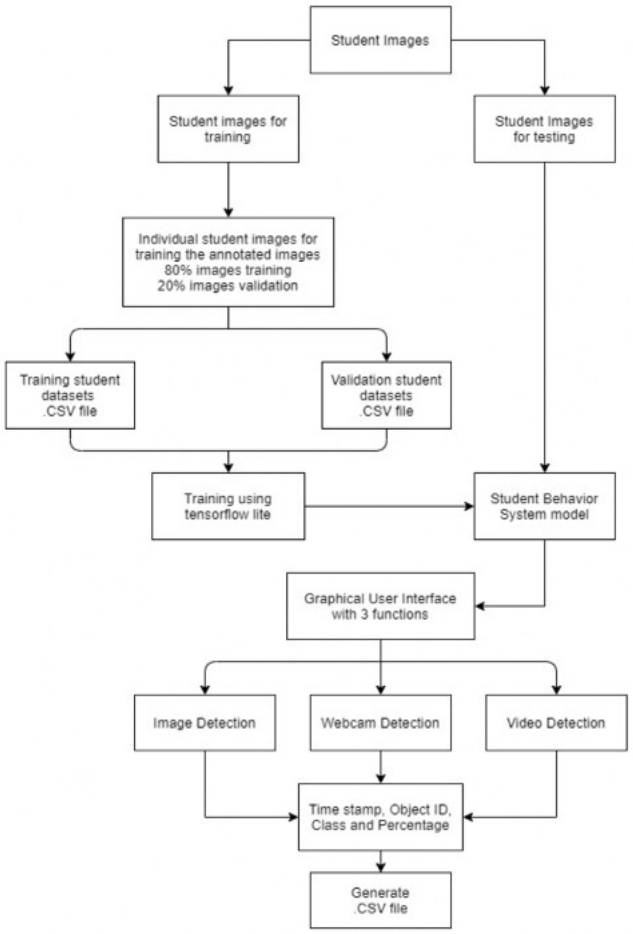

Figure 2. Proposed Methodology

After data collection, the images were prepared for the training and testing phase. The images were annotated and split for the training and validation phase. A simple graphical user interface (GUI) was developed for the prediction of the student behavior based on facial cues. The system generated output was the basis of the detection.

3.2 Dataset and Materials

The following are the materials used in data generation and system development. The system has two main components: hardware, software, and dataset.

A. Hardware Components

The system used Intel Core i7 2.8 GHz with 4.0 GB RAM and NVIDIA GeForce GTX 1050. A webcam was used to capture images and test the model.

B. Software Components

Anaconda Python 3.7 was used for the virtual environment with CUDA and DNN toolkits versions that are compatible with the TensorFlow with GPU. This was run on the Windows 10 operating system. TensorFlow Light algorithm was used to data and training and TensorFlow Lite Optimizing Converter (TOCO) was used to create an optimized model. The system also used MSYS2 and Visual C++ Build Tools 2015 to create a simple GUI for data testing. The GUI has three options whether the data is an imported image, video, or live feed.

C. Dataset

The generation of datasets was accomplished using a webcam. The data was collected in the form of an image. The method used in data collection and processing is shown Figure 3 below.

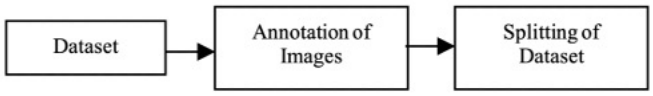

Figure 3: Dataset Generation and Preparation

The data collected is a collection of images that includes different viewpoints on the faces of each student in a single computer lab. A separate collection of data has been obtained to ensure that it is accessible enough to be used for training purposes. Data were manually annotated and split for the training and validation of the model generation dataset.

To order to train a reliable classifier, the training images should have random objects to the image along with the target objects and should have a range of backgrounds and lighting conditions. The label used was labelimg.exe to label the images. This was done in the generate_tfrecord.py file in a text editor.

```
def class_text_to_int(row_label):
    if row_label == 'attentive':
        return 1
    elif row_label == 'non_attentive':
        return 2
else:
        None
```

Figure 4: Creation of label map and configuration

The creation and implementation of the label map is shown in Figure 4. The mark used was both attentive and non-attentive. Attentive means the student is concentrated and alert, while the non-attentive is relaxed. The label map specifies the trainer what each object is by mapping class names to class ID numbers. Subsequently, an object detection training pipeline must be built. This determines which model and which parameters will be used for training.

3.3 Training

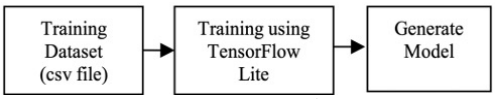

Figure 5: Training of Model

Shown in Figure 5 is the steps of training the model. The system utilized the TensorFlow Lite algorithm for dataset training. Below illustrates the dataset training phase.

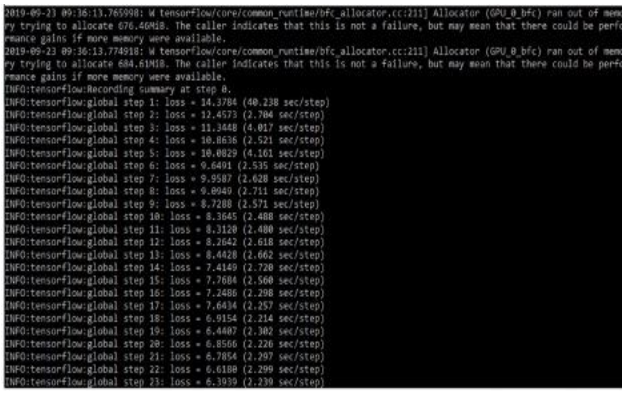

Figure 6: Dataset Training

Figure 6 shows the training of the dataset. The model created was exported for conversion to TensorFlow Lite using the export file ssd graph.py script. Training used a GPU-based computer system to reduce training time. The model was produced after the training. TensorFlow Lite Optimizing Converter (TOCO) was used to create an optimized model as shown in Figure 7. To build the TensorFlow, MSYS2, Visual C++ Build Tools 2015 was established, anaconda, and create a TensorFlow-build environment, Bazel, and Python package dependencies. The value of the average loss was measured to assess if the model is under-fitting or over-fitting.

```
conda install -c menpo wget
wget https://raw.githubusercontent.com/EdjeElectronics/TensorFlow-Lite-Object-
Detection-on-Android-and-Raspberry-Pi/master/TFLite_detection_image.py --no-check-
certificate
wget https://raw.githubusercontent.com/EdjeElectronics/TensorFlow-Lite-Object-
Detection-on-Android-and-Raspberry-Pi/master/TFLite_detection_video.py --no-check-
certificate
wget https://raw.githubusercontent.com/EdjeElectronics/TensorFlow-Lite-Object-
Detection-on-Android-and-Raspberry-Pi/master/TFLite_detection_webcam.py --no-check-
certificate
```

Figure 7: TensorFlow Lite Configuration

3.6 Implementation

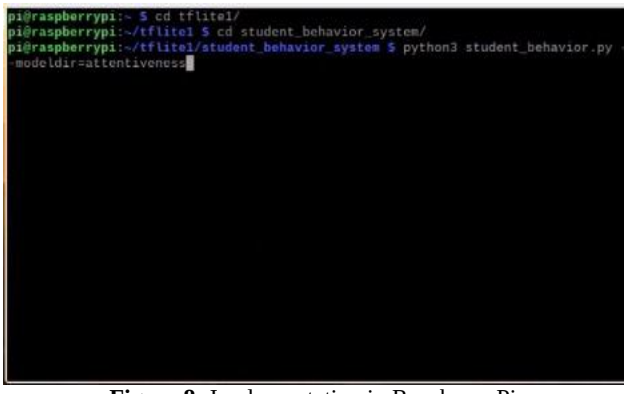

Figure 8: Implementation in Raspberry Pi

The model that was developed was applied to each frame from the capturing device. The model interface was implemented on the raspberry pi desktop as shown in Figure 8. Figure 9 shows a simple GUI was designed to effectively test the functionality of the system.

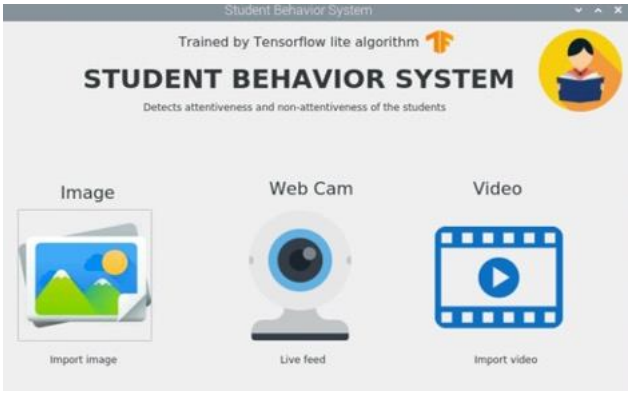

Figure 9: User Design Interface

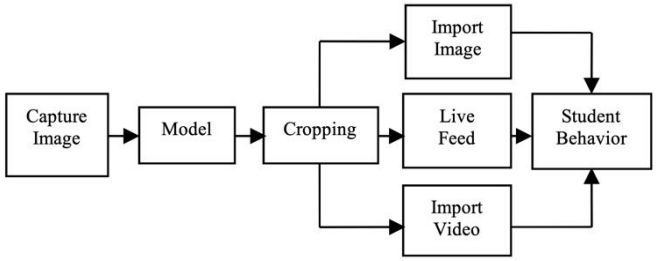

Figure 10: Prediction of Student Behavior

Three options were utilized for the implementation; import image, live feed, and import video as shown in Figure 10. The produced output a CSV file of the student behavior.

4. RESULTS AND DISCUSSIONS

Data collected was carried out in a computer laboratory where students were told to behave appropriately while class discussions were taking place. Image detection, webcam detection, and video detection were performed. Each feature of the built GUI has been checked for the functionality of the system.

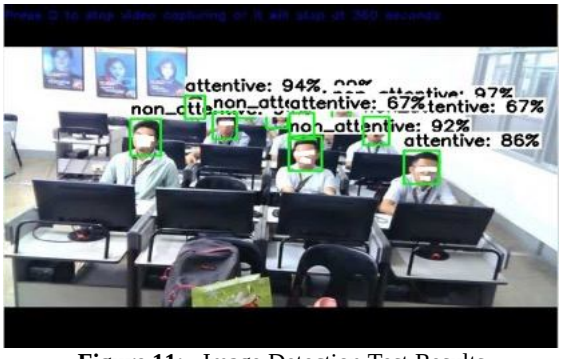

Figure 11: Image Detection Test Results

Shown in Figure 11 is the image detection result of testing. Each student's face was recognized, and student behavior was also detected.

Table 1: Test Results of Face Detection

Input Image	Number of Faces in the Image	Number of Face Recognized	Face Recognition Accuracy
1	9	9	100%
2	9	9	100%
3	9	9	100%
4	9	7	77%
5	9	9	100%
6	9	8	88%
7	9	9	100%
8	9	9	100%
9	9	9	100%
10	9	9	100%

The results of the face detection of the input images are show n in Table 1. Images from input numbers 4 and 6 result in 77% and 88% of facial recognition due to the student's location orientation, which is positioned almost in the exact position of the student in front of the back. As a result, the student's face was not visible to the camera. Because of these, four more images were employed to test the system. After a total of 10 image testing, the overall testing performed of face recognition results to an accuracy of 96.7%

Below are the visuals of the percentage of the actual detection of student attentiveness (student 0 – student 8).

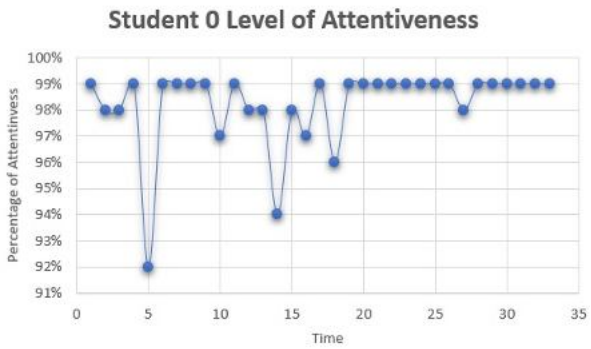

Figure 12: Student 0 Level of Attentiveness Actual Detection

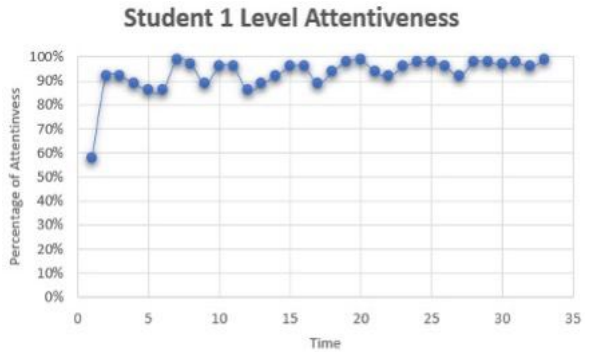

Figure 13: Student 1 Level of Attentiveness Actual Detection

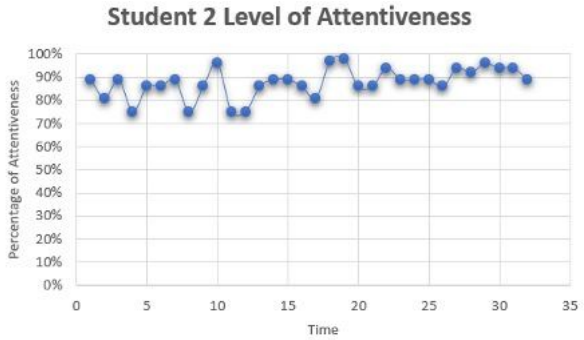

Figure 14: Student 2 Level of Attentiveness Actual Detection

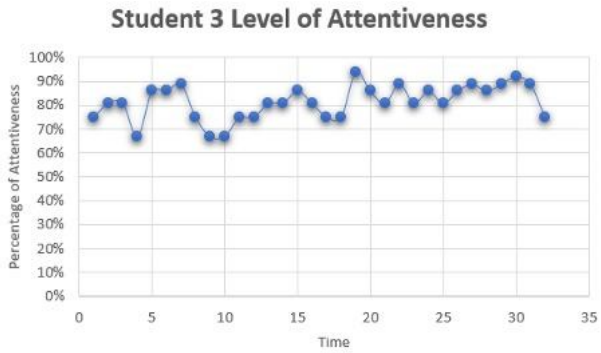

Figure 15: Student 3 Level of Attentiveness Actual Detection

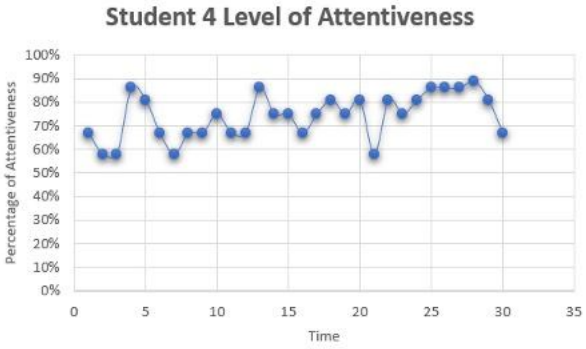

Figure 16: Student 4 Level of Attentiveness Actual Detection

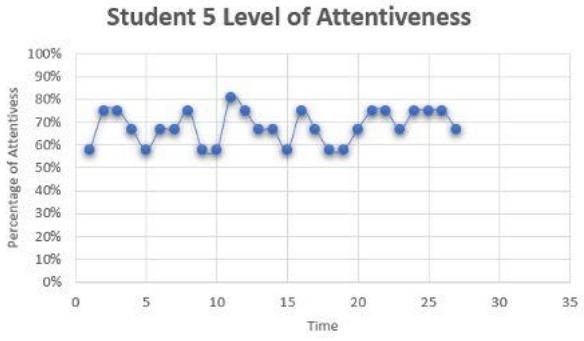

Figure 17: Student 5 Level of Attentiveness Actual Detection

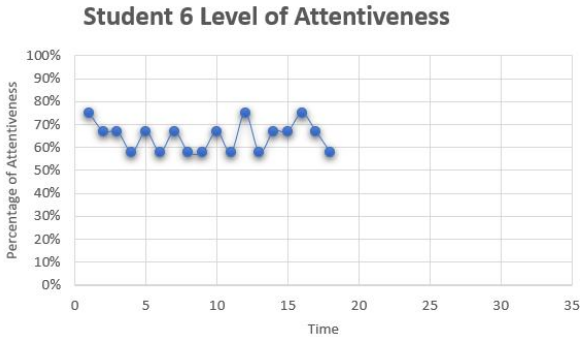

Figure 18: Student 6 Level of Attentiveness Actual Detection

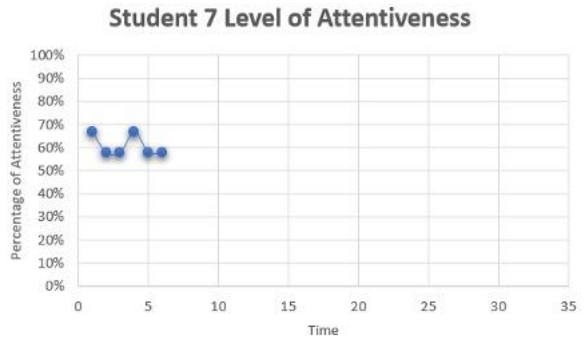

Figure 19: Student 7 Level of Attentiveness Actual Detection

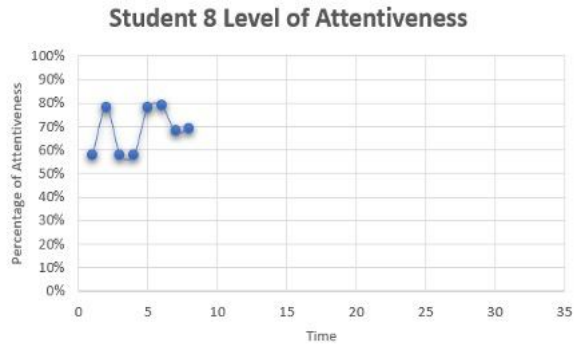

Figure 20: Student 8 Level of Attentiveness Actual Detection

The result of the live feed and video data detection based on the thirty-five second time stamp of the three hours total length of the class discussion as shown in Figures 12 - 20. Every student's face was identified, and every focus was also captured. Percentage of attention of each student: student 0 – student 8 (subject ID) is the actual measurement if the student pays close attention during class discussion. Among students 6, 7 and 8, the findings indicate that these student faces have not been identified at times. Furthermore, the monitoring of student conduct was not determined because of this.

5. CONCLUSION

As shown in the results of the facial recognition, the images in input numbers 4 and 6 result in 77% and 88% facial recognition due to the orientation of the student's location, which is located almost in the exact position of the student in front of the back. As a result, the level of attention to these students was also affected. Students 6,7 and 8 were three students with a low level of concentration from thirty-five seconds of monitoring. To overcome these, additional inputs were used to check the accuracy of the model. Upon further testing, the results show that face recognition results in an accuracy of 96.7%, and attention detection has also been performed. With these results, this indicates that the model generated provides an accurate model. Further class discussions can be carried out using this method. On the other side, a dome style of camera or an IP camera may be used to fix problems in the image capture of the student's position orientation. It can be done for future research.

ACKNOWLEDGEMENT

The authors would like to thank the CISCO 4 class of Computer Engineering students for the data collection and MR. Suave Data Science Laboratory of the Technological Institute of the Philippines – Manila for the computing materials used for the completion of this study.

REFERENCES

1. V. Kassarnig, A. Bjerre-nielsen, E. Mones, S. Lehmann, and D. Lassen, **Class attendance , peer similarity , and academic performance in a large field study,** pp. 1–15, 2017.
2. R. Pekrun, ***Emotions and Learning***. Belley, France: International Buruea of Education, 2014.
3. A. Košir, **Predicting students ' attention in the classroom from Kinect facial and body features,** *J. Image andVideo Process.*, 2017. https://doi.org/10.1186/s13640-017-0228-8
4. C. Thomas and D. B. Jayagopi, **Predicting student engagement in classrooms using facial behavioral cues,** *MIE 2017 - Proc. 1st ACM SIGCHI Int. Work. Multimodal Interact. Educ. Co-located with ICMI 2017*, vol. 2017-Novem, pp. 33–40, 2017.
5. P. Goldberg, Ö. Sümer, K. Stürmer, and W. Wagner, **Attentive or Not ? Toward a Machine Learning Approach to Assessing Students ' Visible Engagement in Classroom Instruction,** 2019.
6. J. Marot and S. Bourennane, **Raspberry Pi for image processing education,** *25th Eur. Signal Process. Conf. EUSIPCO 2017*, vol. 2017-Janua, no. August, pp. 2364–2368, 2017.
7. C. James and D. Nettikadan, **Student monitoring system for school bus using facial recognition,** *Proc. Int. Conf. Trends Electron. Informatics, ICOEI 2019*, vol. 2019-April, no. Icoei, pp. 659–663, 2019. https://doi.org/10.1109/ICOEI.2019.8862534
8. S. Arya, N. Pratap, and K. Bhatia, **Future of Face Recognition : A Review,** *Procedia - Procedia Comput. Sci.*, vol. 58, pp. 578–585, 2015.
9. A. S. Hasban *et al.*, **Face recognition for Student Attendance using Raspberry Pi,** *APACE 2019 - 2019 IEEE Asia-Pacific Conf. Appl. Electromagn. Proc.*, no. November, pp. 25–27, 2019.
10 I. Gupta, V. Patil, C. Kadam, and S. Dumbre, **Face**

Detection and Recognition using Raspberry Pi, *2016 IEEE Int. WIE Conf. Electr. Comput. Eng.*, no. December, pp. 19–21, 2016.

11. S. M. Hatturea, **Home Security against Human Intrusion using Raspberry Pi,** *Procedia Comput. Sci.*, vol. 167, no. Iccids 2019, pp. 1811–1820, 2020.
12. Z. Balogh, M. Magdin, and G. Molnár, **Motion Detection and Face Recognition using Raspberry Pi, as a Part of, the Internet of Things,** no. June, 2019.
13. S. Oltean, **Mobile Robot Platform with Arduino Uno and Raspberry Pi for Autonomous Navigation,** *Procedia Manuf.*, vol. 32, pp. 572–577, 2019.
14 S. Wisnusenna and M. Sunardy, **Model of Web Based Application to Control Bridge Traveler Using Raspberry Pi,** *Procedia Comput. Sci.*, vol. 135, pp. 518–525, 2018.
15. V. Pasquali, G. D. Alessandro, R. Gualtieri, and F. Leccese, **A new data logger based on Raspberry-Pi for Arctic Notostraca locomotion investigations,** *Measurement*, vol. 110, pp. 249–256, 2017.
16. S. W. Foster, M. J. Alirangues, J. A. Naese, E. Constans, and J. P. Grinias, **Short communication A low-cost , open-source digital stripchart recorder for chromatographic detectors using a Raspberry Pi,** *J. Chromatogr. A*, vol. 1603, pp. 396–400, 2019. https://doi.org/10.1016/j.chroma.2019.03.070
17. A. Cempaka, A. Rahayu, and W. Budiharto, **Developing Information System of Attendance and Facebook Status f or Binus University ' s Lecturer Using Raspberry Pi Architecture,** *Procedia - Procedia Comput. Sci.*, vol. 59, no. Iccsci, pp. 178–187, 2015.
18. J. G. Nicolas-Mindoro, M. A. F. Malbog, J. S. Gulmatico, and J. B. Enriquez, **Notipoy: Early notification of detected fire using image processing applying convolutional neural network,** *Int. J. Adv. Trends Comput. Sci. Eng.*, vol. 8, no. 6, pp. 2811–2816, 2019. https://doi.org/10.30534/ijatcse/2019/21862019
19. S. Athi Narayanan, M. R. Kaimal, K. Bijlani, M. Prasanth, and K. Sunil Kumar, **Computer vision based attentiveness detection methods in E-Learning,** *ACM Int. Conf. Proceeding Ser.*, vol. 10-11-Octo, no. August 2016, 2014.
20. A. S. Alon, C. D. Casuat, M. A. F. Malbog, R. I. Marasigan, and J. S. Gulmatico, **A YOLOv3 inference approach for student attendance face recognition system,** *Int. J. Emerg. Trends Eng. Res.*, vol. 8, no. 2, pp. 384–390, 2020. https://doi.org/10.30534/ijeter/2020/24822020
21. P. Lin, Q. Li, Q. Fan, and X. Gao, **Real-Time Monitoring System for Workers ' Behaviour Analysis on a Large-Dam Construction Site,** vol. 2013, 2013.
22. M. A. Malbog, **MASK R-CNN for Pedestrian Crosswalk Detection and Instance Segmentation**, *2019 IEEE 6th International Conference on Engineering Technologies and Applied Sciences (ICETAS)*, Kuala Lumpur, Malaysia, 2019, pp. 1-5, doi: 10.1109/ICETAS48360.2019.9117217.