Under the supervision of Prof. Samah El-Tantawy

Detection of Student Disengagement in Online Classes Using Deep Learning

Shahd Ahmed, Meram Mahmoud, Nouran Hani, Mostafa Ali, Ahmed Mohamed, Mohamed Hisham

Department of systems and Biomedical Engineering (SBME),
Cairo University
Partial Differential Equations (PDEs) and Special Functions [MTH2245]

Keywords: "Online learning", "Disengagement Detection", "Facial Landmarks", "Drowsiness", "Student disengagement", "Computer vision", "Deep learning", "CNN"

II. ABSTRACT

Online learning has become the new norm of education especially after the closure of the educational institutions during the Covid-19 pandemic. However, online learning faces different challenges, one of which is the lack of student participation in classes leading to disengage in the session and drop out from the learning process. This disengagement, not easily noticed by teachers, significantly impacts teaching effectiveness. This research proposes a computer vision solution which tracks and assesses student engagement in real-time. Focusing on the use of deep learning methods in analyzing facial expressions and drowsiness indicators to detect signs of disengagement such as yawning and drowsiness. This study compares two appearance-based models: VGG16 transfer learning model and the Facial Landmarks based neural network. Both models showed acceptable accuracy; however, the VGG16 transfer learning model showed superior performance and accuracy.

II. Introduction and Problem Definition

Online learning has become the new norm of education specially after the closure of the educational institutions during the Covid-19 pandemic which affected 94% of students in 200 countries according to UN.[1] The closure followed by a huge transformation to virtual classes and online learning Figure 1 which has appeared in the dramatic growth of MOOC's (Massive Open Online Courses) platforms like Coursera and video conference sites like ZOOM.[2] Numbers of learners enrolled in distance education courses have dramatically increased. In fall 2021, 60 percent of total USA students have enrolled in at least one e-learning course. [3]

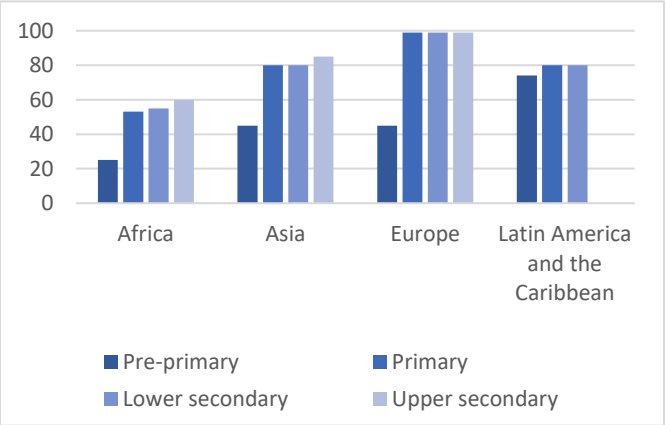


Figure 1Country Choice of Online Learning During School Closure

One of the challenges that faces online learning is the lack of student's participation in the classes which is part of learning disengagement.[4] That is highlighted in a survey was conducted on students during the shift to online learning in the closure period. Nearly, 52% of students have agreed that their understanding of the material has decreased Figure 2. At the same time, 61% of the students have agreed that online classes decreased their desire to participate and engage in classes Figure 3.[5]

Engagement in the context of education has several definitions, all of them agree that engagement includes the emotional and attentional involvement within a task.[6] Based on multiple researchers, engaging students in the learning material helps in improving the understanding of concepts and absorption of the material. For instance, a study conducted on high school students highlights that there is a direct positive relation between students' engagement and academic performance, as the more the student is engaged in the study material the higher grades they get[7].

Teachers face challenges in tracking their students' engagement due to the complexity of the online learning environment. Additionally, when the course material is prerecorded, teachers may not have direct access to students which results in their disengagement. The continuity of student disengagement can lead to totally drop out from the course.[4] Assessing and measuring the student engagement during the class is a key factor in solving the problem of disengagement as it allows the educators to track their student's attention levels which is especially important in online settings.[8]

This paper is proposing a computer vision (CV) and deep learning (DL) solution to automate the process of assessing and tracking students' engagement in online settings by tracking the common behaviors of disengaged students like: Yawning, Drowsiness.

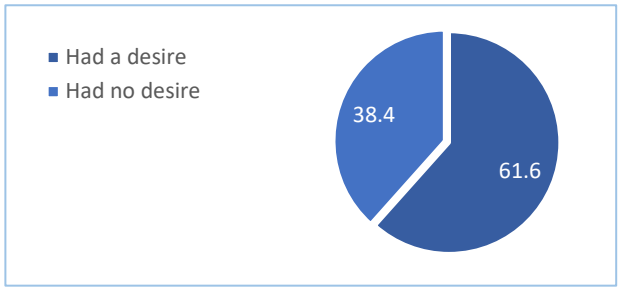


Figure 2 Students' opinion on how the shifting to online classes has affected their understanding.

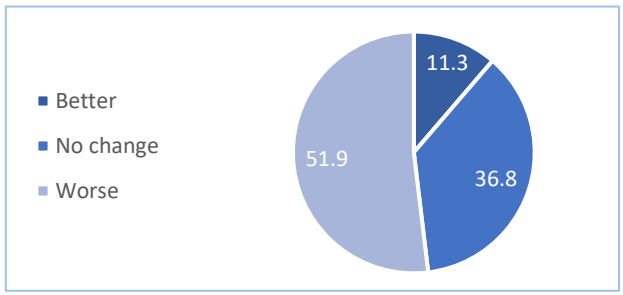


Figure 3 Students' opinion about how online learning affected their desire to participate.

III. Litrature Review

Student's Engagement was first introduced in the educational context in the 1980s, to understand and address issues such as student boredom and dropout rates, as discussed in Newmann's study that conceptualized student engagement. One of the challenges is the lack of students' participation in the classes which is part of learning disengagement.[7]

Axelson and Flick defined Student engagement as: **"*How involved or interested students appear to be in their learning and how connected they are to their classes, their institutions, and each other*"**[9]. This concept has been supported by a survey conducted on high school students, which found a direct positive relationship between student engagement and academic performance, as the more the student engaged in the study material, the higher grades they get. [10]

As the definition of engagement has become more complex, it has been divided into various components. One model, according to Fredric Categorization of learners engagement[11], divides engagement into three components: behavioral, cognitive, and emotional. Similarly, Bosch divides engagement into three categories: affective, behavioral, and cognitive.[8] In contrast, Anderson's research expanded the components of engagement to include behavioral, cognitive, academic, and psychological dimensions [12]

In conclusion, based on several readings, learner engagement can be defined as the feelings that drive actions reflecting a student's interest in academic material. These actions include both individual and peer activities, occurring in both classroom settings and extracurricular activities.

Now, Online learning faces different challenges, one of which is the lack of student participation in classes leading to disengage in the session and drop out from the learning process. [2] This lack of engagement was observed during the shifting to online education in response to the COVID-19 pandemic when online learning had become the new norm of education especially after the closure of educational institutions in over 200 countries.[1]

Assessing and measuring student engagement during class is a key factor in solving the problem of disengagement, particularly in online settings. It enables educators to track the attention levels of their students [6]. Methods for measuring student engagement can be categorized into three main types based on the level of learner involvement in the detection process: automatic, semi-automatic, and manual.[13]

In our literature, we are concerned about the automatic assessment of student engagement using Artificial Intelligence (AI) technologies, with a focus on Deep Learning models.

We have collected relevant studies after using this search string - **("learners" OR "students") AND ("online learning" OR "online education") AND ("Engagement Assessment" OR "Engagement Tracking" OR "Disengagement Detection") AND ("computer vision" OR "Deep learning")** -which was created and subsequently applied to the Google Scholar database for searching titles and abstracts to find relevant articles.

In order to set our inclusion and exclusion criteria[1] for study selection, we have followed Cooper's guidelines, [14] :

Inclusion criteria:

- **Language**: Studies must be written in English.
- **Study type:** Accepted study types include empirical studies (full articles, papers, notes, extended abstracts, and work-in-progress papers).
- **Peer-Reviewed:** Studies should have undergone peer review.
- **Focus:** Studies must exclusively focus on Online Learning.
- **Length**: A minimum of four pages is required.
- **Engagement Focus:** Research must explicitly focus on learners' engagement in Online Learning, offering insights into enhancing engagement or deepening the understanding of the topic.
- **Publication date:** Studies published between January 1, 2013, and November 1, 2023, are eligible.
- **No Replication:** Studies should not duplicate the same idea by the same author (s).
- **Source Types:** Both journal articles and papers included in conference proceedings are acceptable.

Exclusion criteria:

- **Non-English Language:** Studies not written in English are excluded.
- **Irrelevant Source Types:** Blog posts, magazine articles, theses, newsletters, and literature review articles or papers do not meet the inclusion criteria.
- **Repetition by Same Authors:** Repeated contributions by the same authors in journal articles and conference papers are not eligible.

The search process initially identified 272 studies, but after applying our exclusion and inclusion criteria, only 72 study met our criteria. We have established a CSV [2] spreadsheet to systematically gather excerpts from all these studies, organizing the information in rows and columns. Each row provided a summary of the data extracted from each study, while the columns detailed the types of data being extracted. These columns typically included information such as the title, author, publication year, article or paper type, and other relevant data. Furthermore, we have chosen out of the 72 study only 38 study[3] after reviewing the results once more. Figure 4

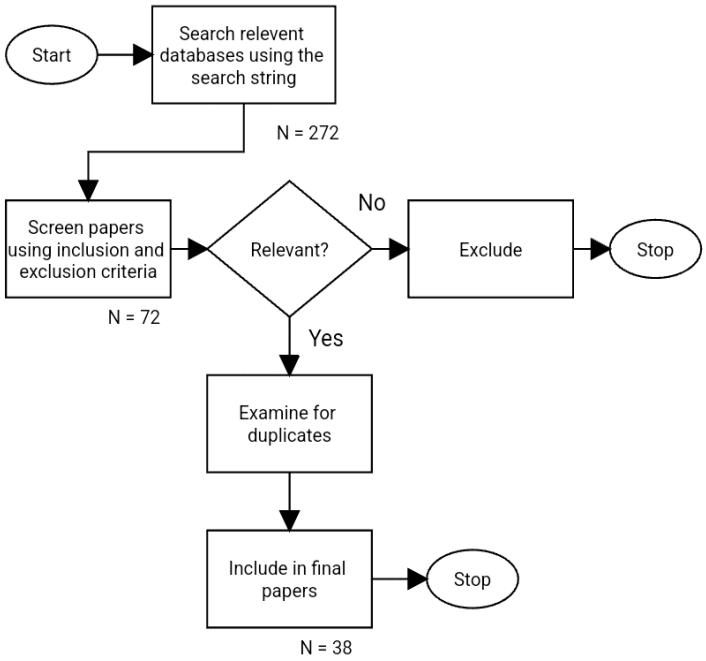


Figure 4 Studies selection process

In our research, we are concerned about utilizing AI for assessing engagement, particularly focusing on computer vision techniques due to their ease of use and the widespread availability of low-cost cameras. Additionally, we are exploring the use of affective computing techniques, which closely resemble teacher observations and do not disrupt the learning process for the students during the session.

As stated, 38 papers from our results were found that depend on computer vision. These papers presented various approaches for detecting disengagement, along with different methods and models, which will be discussed below.

We examined the collected research to gather answers to our research questions, which are as follows:

1. Which methods or technologies are used to assess students' engagement?

[2] All the 72 research with details: Selected Studies details
[3] Detailed description of our 38 main review studies: Link

2. Which indicators or specific acts are assessed to detect disengagement?

Additionally, we gathered demographic information about the studies. Studies were published from 2014 to the present date, with the year 2023 having the highest publications count Figure 7. The major focus of these studies was on students, with sample sizes ranging from 19 to 432 students. Their ages vary between 17 to 29 years. A detailed description of every paper sample was added to the studies data sheet mentioned earlier.

Computer vision-based models offer various ways to assess learners' engagement, with the most common modalities being facial expressions, gestures, postures, and eye movements. Out of 37 papers investigated, 21 of them focused on facial expressions Figure 5. Therefore, we can categorize the approaches into two main groups: Face-dependent and Face-independent Figure 6.

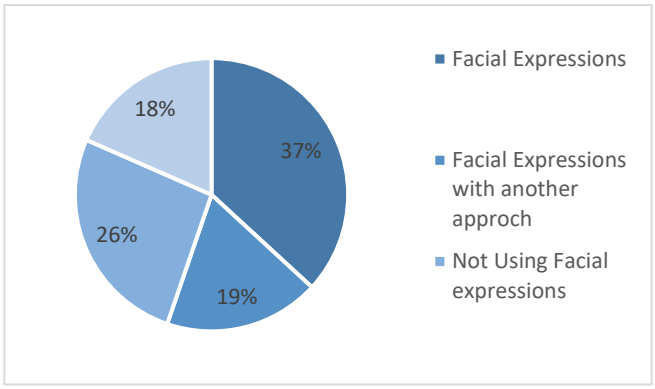


Figure 5 Engagement Assessments Approaches Found in Review Papers

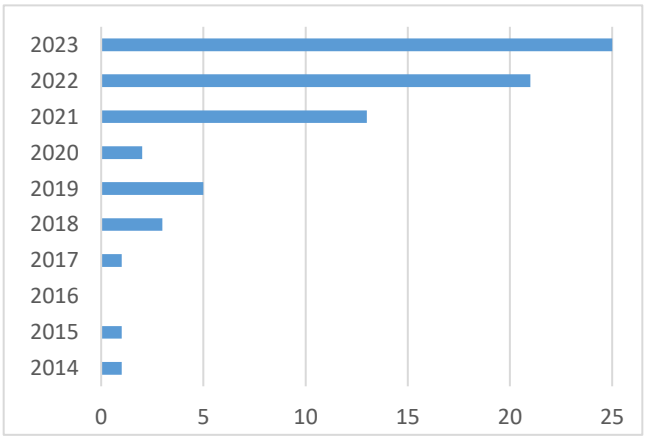


Figure 7 Numbers of Publications Found Every Year

Face-dependent assessments can be further divided into two types: part-based indicators and appearance-based indicators. Part-based indicators concentrate on specific facial features such as eyes and mouth. The Facial Action Coding System (FACS) is utilized to analyze different facial parts, referred to as Action Units (AUs), which represent the movements of facial muscles. These units are used to measure specific emotions [12]. These emotions then reflect the level of student engagement with the learning material.

Appearance-based models, on the other hand, rely on extracting features from the entire face and generating patterns for engagement classification. In [15], a lightweight attentional convolutional neural network (CNN) is introduced for face expression recognition. This model recognizes four main expressions, each of which is assigned a weight and contributes to the assessment of engagement based on specific thresholds and equations, dependent on a trained CNN model.

In our literature, approximately 10 studies have focused on indicators that are not directly related to facial expressions but rather on user activities during the session, such as mouse activity or user posture. In [16] and [17], learner's mouse activity is integrated with their gaze during the session and used to train AI models for detecting learner disengagement in online settings. The first study employed a CNN model trained to establish a connection between mouse activity and gaze, while the other study used data from a trained Support Vector Machine (SVM) model to analyze user mouse activity in a learning session to detect disengagement. They achieved this using their dataset of computer users' mouse activities during specific tasks.

In another study [18], the activity data of 360 students within the e-learning platform was collected and summarized into eight features, which included Total Logins, Activity inside the content area, Number of Clicks, Join Session, User Activity Group, Time Spent, Total Items, and Time Spent in Session Attendance. Subsequently, this labeled data was used as input for two models: an SVM model and an Artificial Neural Network.

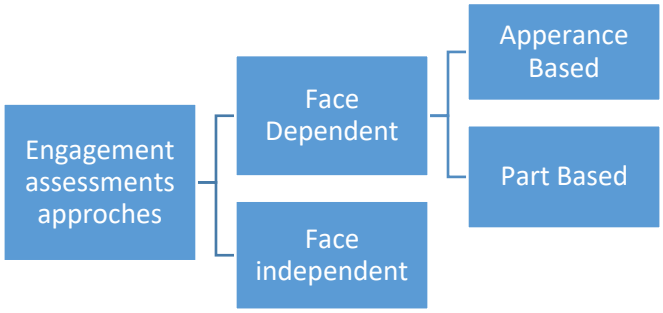


Figure 6 Engagement Assessment Approaches

Based on our review, we have not found yet any studies that specifically illustrate the level of engagement of learners based on their drowsiness status. As mentioned earlier [10], drowsiness is considered one of the most significant indicators of disengagement during online learning. Using various deep learning models, our goal is to introduce an innovative framework for assessing student engagement by considering drowsiness status. This model is proposed as a web app solution for instructors and education providers, which will assist them in continuously monitoring their learners' levels of engagement.

IV. METHODS

A. Convolutional Neural Networks (CNN)

Convolutional Neural Network (CNN) is a cutting-edge deep learning network that imitates how the visual cortex of the brain processes and recognizes images, which is valuable for identifying patterns in data.[15] It typically comprises three layers that perform convolution operations on the input data: a convolutional layer, a pooling layer, and a fully connected (FC) layer, allowing the network to learn and extract features from images or videos.[16]

The convolutional layer is the core building block of CNNs, carrying the main computational load of the network. It possesses the characteristics of local connection and value sharing.[17] After performing the convolution operation with a filter, it generates new images known as "feature maps" which offers better representation of features.[15]

The number of channels must be the same for the input image and the filter. As each filter is applied to each channel, the results are combined into a representative value. The number of output channels is equal to the number of filters applied. Figure 10

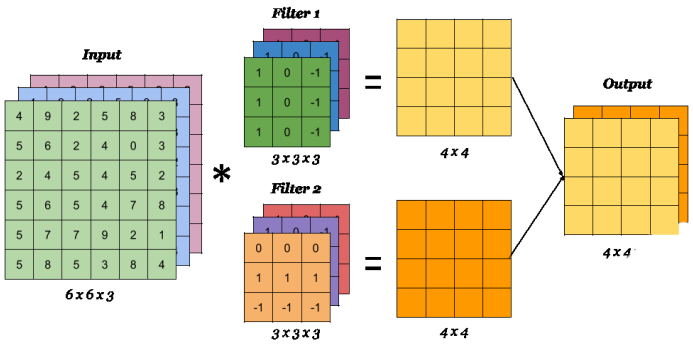


Figure 10 Two filters of three channels

The kernel, also known as a “feature detector” or a “filter” (f), is used for feature extraction from data by sliding over the input data (n) and performing convolution, the sum of multiple element-wise operations on sub-regions. The output from this operation is the feature map. Figure 11 [15]

$$output\ size = (n - f + 1) \times (n - f + 1) \qquad (1)$$

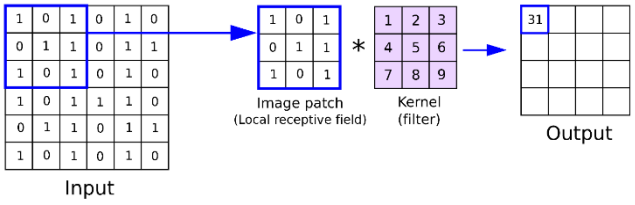


Figure 11 Process of applying Kernal on the image.

The pooling layer is responsible for reducing the dimensionality of the output image, retaining only the important features. It also has the capability to mitigate the overfitting issue, making it beneficial for computational efficiency [15]. Pooling is applied similarly to the convolution operator by sliding a kernel of size (f) across the input image, with a commonly used stride (s) of 2 and almost no padding $p = 0$.

Pooling is divided into two types: max pooling, and average pooling. [Figure 2] In Max Pooling, which is the more prevalent one, the operation involves preserving the maximum value of each patch, while in Average Pooling, it preserves the average. Figure 8 [18]

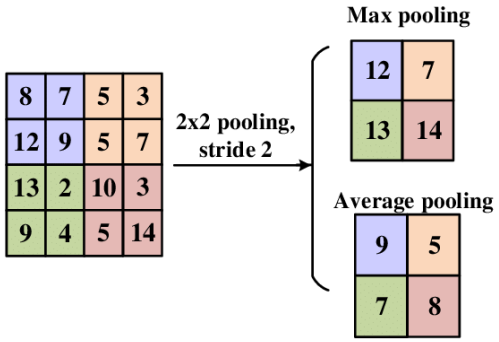


Figure 8 Max pooling vs Average pooling

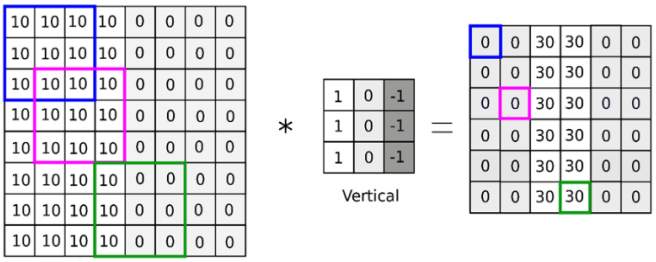


Figure 9 Example of vertical edge detection

Edge detection involves identifying object boundaries within a still image using a filter that detects sudden changes in colors or intensity within the regions of the input image. For instance, it can detect the vertical edge Figure 9, illustrated by the brighter region in the middle of the output feature map, which indicates a sudden change in the middle of the input image.[17]

The resulting image from the filter shrinks after each convolution operation because that borders’ pixels of the original image carry less weight in the convolution operation compared to center’s pixels. Consequently, valuable information are lost in this process.[16] To address this, padding (p) involves adding extra layer(s) of zeros to the borders of the original image in every direction.[18]

The dimensions of the input image become $(n + 2p) \times (n + 2p)$ and the dimensions of the output image become $(n + 2p - f + 1) \times (n + 2p - f + 1)$. When making no padding ($p = 0$), this is called "valid" convolution. When $p = \frac{f-1}{2}$ the size of the output will be the same as input, which is called "same" convolution.

Stride (s) is a hyperparameter of the neural network’s kernel that determines the movement of the filter over the image.[18] A stride of 1 implies that the filter will move pixel by pixel. Similarly, setting it to 2 means it will skip 2 pixels in the sliding process.

$$output\ size = (\frac{n+2p-f}{s} + 1)(\frac{n+2p-f}{s} + 1) \qquad (2)$$

B. *Forward Propagation:*

Forward propagation (forward pass) is one of the core processes during the learning phase It involves computing and storing intermediate variables, including outputs, within a neural network, progressing sequentially from the input layer to the output layer.[19] Each hidden layer receives the input data, processes it based on the activation function, and then passes it to the next layer.

At each neuron in the hidden or output layers, the processing goes throw two phases:

1- **Pre-activation:** This phase involves computing a weighted sum of inputs. Before deciding, a neuron accumulates the information received, assigning different weights to each piece of information (input) based on its significance.

2- **Activation:** the calculated weighted sum is passed to the activation function, a mathematical function which introduces nonlinearity to the network.[20] Examples of activation functions are sigmoid, hyperbolic tangent (tanh), Rectified Linear Unit (ReLU) Figure 12, and Softmax Figure 13.

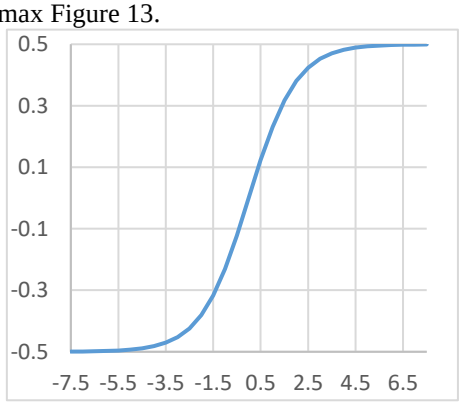


Figure 13 Softmax Activation Function Graph

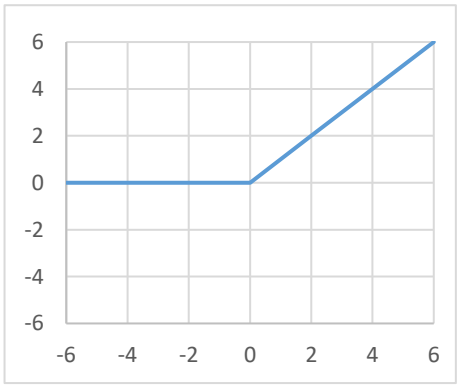


Figure 12 ReLu Activation Function

Forward propagation is applied in all CNN layers, but it differs a little based on the layer type through which is applied. The forward propagation through the convolutional layers is contrasted by the below equation as it shows the output of the pre-activation for a $N \times N$ square neuron layer using a $m \times m$ filter ω.

$$x_{ij}^{l} = \sum_{a=0}^{m-1} \sum_{b=0}^{m-1} w_{ab} y_{(i+a)(j+b)}^{l-1} + b^{l} \quad (3)$$

After computing x_{ij}^{l} the convolutional layer applies its nonlinearity activation function:

$$y_{ij}^{l} = g\left(x_{ij}^{l}\right) \quad (4)$$

Then the pooling layer forward propagation just applying either the max or average pooling on specific region $(K \times K)$ of a $N \times N$ layer which outputs x_{ijc}^{l} of size $\frac{N}{k} \times \frac{N}{k}$.

$$x_{ijc}^{l} = Max\left(x_{i:i+f,j:j+f,c}^{l-1}\right) \quad (5)$$

$$x_{ijc}^{l} = Avg\left(x_{i:i+f,j:j+f,c}^{l-1}\right) \quad (6)$$

The forward propagation in the Fully connected layer can be described by:

$$z^{i[l]} = W^{[l]} a^{(i)[l-1]} + b^{[l]} \quad (7)$$

$$a^{i[l]} = g\left(z^{i[l]}\right) \quad (8)$$

In order to simplfy the idea fo forward propagation, we consider, a multiclass classifier consists of a neural network with three layers where a ReLU activation function connects the input and two hidden layers $a^{(i)[1]}$ $a^{(i)[2]}$, and a softmax function connects the final hidden layer and the output layer $a^{(i)[3]} = \hat{y}^{(i)}$, our goal is to predict the class to which the input belongs. Each training example is represented as (x, y), where $x \in \mathbb{R}^{n_x}$ and $y \in \{1, 0\}$, with m being the number of training examples $(x^{(m)}, y^{(m)})$.[21]

Consider the following sequence of operations in a neural network: $x^{(i)} \rightarrow a^{(i)[1]} \rightarrow a^{(i)[2]} \rightarrow \hat{y}^{(i)}$, to compute $a^{(i)[1]}$, we multiply each element from $x^{(i)}$ by the corresponding weight. For instance, the weight $W_{13}^{[1]}$ is multiplied by the third element in $x^{(i)}$ to obtain the first element in the first layer.

The weights associated with $a^{(i)[1]}$ in our diagram are represented by the matrix $W^{[i]}$

$$W^{[1]} = \begin{bmatrix} w_{11}^{[1]} & w_{12}^{[1]} & w_{13}^{[1]} \\ w_{21}^{[1]} & w_{22}^{[1]} & w_{23}^{[1]} \\ w_{31}^{[1]} & w_{32}^{[1]} & w_{33}^{[1]} \\ w_{31}^{[1]} & w_{42}^{[1]} & w_{43}^{[1]} \end{bmatrix}$$

To calculate $a^{(i)[1]}$, we only consider the first row of this matrix, denoted as $W_{1-}^{[1]} = [\, w_{11}^{[1]} \; w_{12}^{[1]} \; w_{13}^{[1]} \,]$. this process involves multiplying each element in $W_{1}^{[1]}$ by the corresponding element in $x^{(1)}$and summing the results. Finally, we add a bias $b_{11}^{[1]}$ to obtain $a^{(i)[1]}$. Figure 14

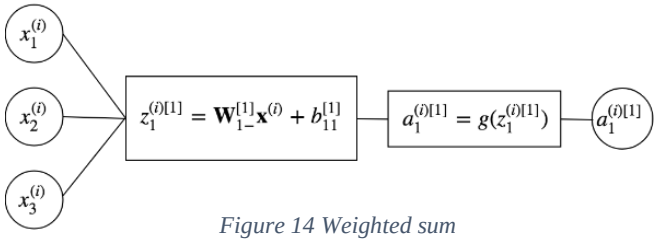


Figure 14 Weighted sum

The activation function $g(z)$ used in the hidden layers is ReLU function [Figure 9], which is defined as [22]:

$$g(z) = \begin{cases} z \;\; if \; z > 0 \\ 0 \;\; if \; z < 0 \end{cases} \quad (9)$$

The forward propagation process is divided into three phases:

Input → 1st Hidden Layer:

1- Activation calculation: dot product of the input features $x^{(i)}$ and the weights $W^{[1]}$ then add the bias $b^{[1]}$ to get $z^{(i)[1]}$

$$z^{(i)[1]} = W^{[1]}x^{(i)} + b^{[1]}$$

2- Apply ReLU activation function element wise to $z^{(i)[1]}$ to get $a^{(i)[1]} = g\left(z^{(i)[1]}\right)$ where $a^{(i)[1]}$ has dimensions (4,1).

1st Hidden Layer → 2nd Hidden Layer:

1- Activation calculation: dot product of $a^{(i)[1]}$ and the weights $W^{[2]}$ and then the bias $b^{[2]}$is added to get $z^{(i)[2]}$

$z^{(i)[2]} = W^{[2]}a^{(i)[1]} + b^{[2]}$

2- Applying the activation function: Apply the (ReLU) activation to $z^{(i)[2]}$ to get $a^{(i)[2]} = g(z^{(i)[2]})$ where $a^{(i)[2]}$ has dimensions (2,1).

2nd Hidden Layer → Output:

1- Activation calculation: dot product of $a^{(i)[2]}$ by the weights $W^{[3]}$ and the bias $b^{[3]}$ is added to get $z^{(i)[3]}$

$$z^{(i)[3]} = W^{[2]}a^{(i)[3]} + b^{[3]}$$

2- Applying the sigmoid activation function: The sigmoid activation function is rarely used in modern neural networks because it suffers from the vanishing gradient problem, but it is often used as the final activation function before the output as it can squash values to be between 0 and 1.[23] So, it's applied to $z^{(i)[3]}$ to get $y^{(i)}$.

$$y^{(i)} = a^{(i)[3]} = \sigma(z^{(i)[3]})$$

where σ is the sigmoid function and $y^{(i)}$ is the final output.

*C. **Backpropagation**:*

Backpropagation is a technique used to compute the gradient of neural network parameters. This method involves traversing the network in reverse, moving from the output layer back to the input layer, following the chain rule from calculus. The algorithm stores any required partial derivatives.

while calculating the gradient with respect to certain parameters.[19] For simplicity, a network with a single hidden layer is used Figure [11], containing a single hidden unit. The outputs $a^{[1]}$ feed into the output layer, providing the final prediction, and throughout, ReLU activation functions are employed. To better comprehend backpropagation, it is obligatory to understand how a computational graph works.

A computational graph [Figure 16] is a directed graph in which nodes correspond to operations or variables. Variables can supply their values to operations, and operations can feed their outputs into other operations. This way, every node in the graph defines a function of variables, in summary.

Firstly, compute the cost function, a mathematical function compares between the predicted value and the actual value (ground truth) for each element in each set of inputs. The optimum goal is to minimize this cost function. There are various mathematical forms of cost function, but the categorical cross entropy cost function is effective with the multiclassification neural networks.

$$J = -\sum_{i=1}^{c} y_i . log\left(\frac{e^{z_i}}{\sum_{i=1}^{n} e^{z_i}}\right) \quad (10)$$

Then, Applying the partial differentiation of the cost with respect to the output y_i then following the chain rule of calculus to get the derivates of the cost function with respect to the weights and biases of the neurons.

Now, once all the derivatives are found, weights could be updates and an optimization algorithm could be used (gradient descent, Adam algorithm)

$$w_{ab}^{l} = w_{ab}^{l} - \propto \frac{\partial J}{\partial w} \quad (11)$$

$$b^{l} = b^{l} - \alpha \frac{\partial J}{\partial b^{l}} \quad (12)$$

where $\propto$ is the learning rate.

Equation 16 and 17 show how the chain rule is applied in fully connected layer while equation 18 and 19 show its application in convolutional layer.

$$\frac{\partial J}{\partial w_{ab}^{l}} = \frac{\partial J}{\partial y_i} . \frac{\partial y_i}{\partial w_{ab}^{l}} \quad (13)$$

$$\frac{\partial J}{\partial b^{l}} = \frac{\partial J}{\partial y_i} . \frac{\partial y_i}{\partial b^{l}} \quad (14)$$

$$\frac{\partial J}{\partial w_{ab}^{l}} = \sum_{i=0}^{N-f} \sum_{j=0}^{N-f} \frac{\partial J}{\partial x_{ij}^{l}} \frac{\partial x_{ij}^{l}}{\partial w_{ab}^{l}} \quad (15)$$

$$\frac{\partial J}{\partial b^{l}} = \sum_{i=1}^{m} \frac{\partial J}{\partial x_{ab}^{l}} \frac{\partial x_{ab}^{l}}{\partial b^{l}} \quad (16)$$

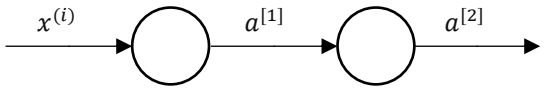


Figure 15 a network with a single hidden layer and two nodes

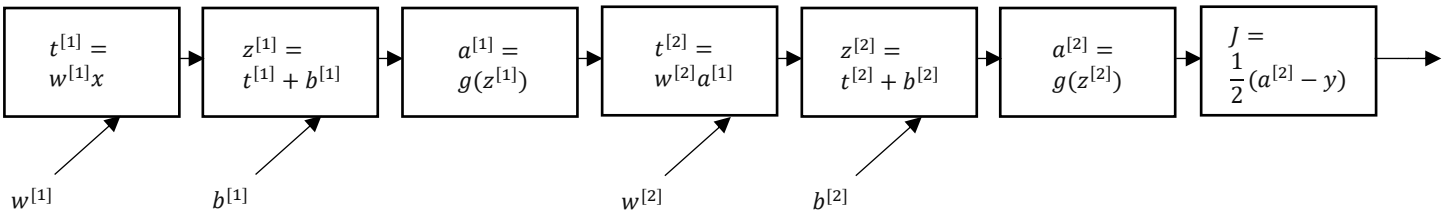


Figure 16 Computational graph

*D. **VGG16**:*

VGG16 is a pre-trained CNN model that was initially introduced at the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) in 2014 by Karen Simonyan & Andrew Zisserman. Their team secured second-place in the classification track using this model. VGG16 stands for Visual Geometry Group, a research group at the University of Oxford. The number 16 refers to the total number of weighted layers.[24]

Input training images must have a fixed size of 224×224 RGB. The model uses 16 convolutional layers with the smallest possible receptive fields of 3×3 and 1×1 convolution filters. In addition to these layers, there are three fully connected layers: the first two layers consist of 4096 channels, and the last one comprises 1000 channels, providing classifications for up to 1000 classes. [Figure 18 All these layers use ReLU activation, and max pooling layers are inserted between them. The dataset used to train this model was the ILSVRC 2012–2014 challenge, which includes 1000 classes. The dataset was split into three sets: Training (1.3 million images), Validation (50,000 images), and Testing (100,000 images), achieving an accuracy of 93%, the best result in terms of single net performance.[24]

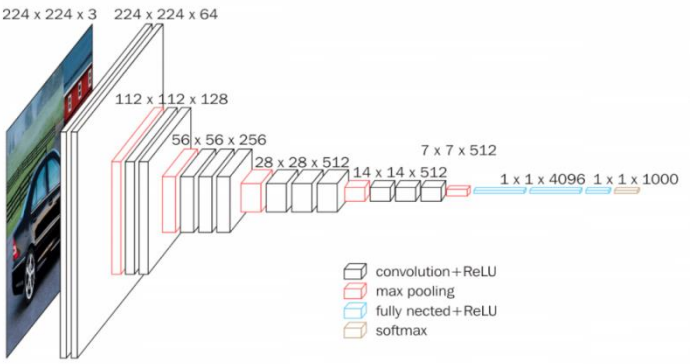


Figure 18 VGG16 Archticture

Transfer Learning (TL) was applied to this model to adapt to project requirements. TL idea was proposed in 1976 by Bozinovski and Fulgosi. Traditional machine learning methods rely on using large datasets to train a model, but due to limited data and the difficulty of gathering more, TL is applied to a compiled dataset. TL finds relations between a pre-trained model and the target model, so the target model needn't to be trained again, resulting in better and faster results. Figure 17 illustrates the difference between traditional learning methods and transfer learning.[25]

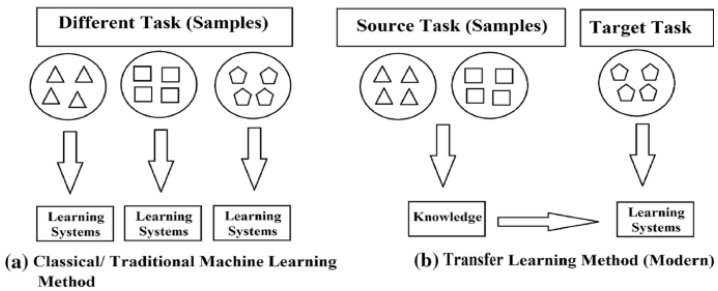


Figure 17 Learning in Transfer Learning Vs Traditional Machine Learning

V. EXPERMENTAL WORK

We have gleaned from our literature review that the two most prevalent approaches for detecting drowsiness as an indicator of student engagement are the Appearance-based method and Part-based methods. In our study, we concentrated on comparing two common methods of appearance-based models to determine which one to implement in our student engagement system. These methods are the CNN transfer learning models and the Facial Landmarks neural network.

A key difference between the two methods lies in how features are extracted from the faces. As discussed earlier, the VGG16 model's pooling layers extract features by capturing hierarchical elements such as edges, textures, and shapes. In contrast, the Face Landmarks method derives features from the Euclidean distances between the landmark points.

*A. **Dataset**:*

In this study, a multifaceted approach was employed for data collection, considering diverse demographic categories such as individuals with black features, narrow eyes, those wearing hijab, and individuals with glasses, spanning ages from 18 to 40 years old. The dataset is organized into three primary categories to ensure a comprehensive and varied dataset for both model training and validation. The first category comprises three open-source datasets. The second category involves images sourced from various open-source image databases. The third category encompasses dataset built by the authors of this paper. The dataset includes images for Egyptian college students from faculty of engineering of Cairo university.

Subsequently, the data was further organized into three sets: training, validation, and test sets. Each set underwent a three-class classification, labeling instances as Active (0), Sleeping (1), and Drowsy (Yawning) (2), to facilitate model handling and interpretation.

Here's a detailed description for the three primary categories:

1) |Opensource datasets:

This category involves three open-source datasets sourced from different research.

Here's a brief description for each dataset:

i) Driver Drowsiness Dataset (DDD):[26] This dataset is derived from extracted and cropped faces of drivers featured in videos from the Real-Life Drowsiness Dataset. VLC software was employed to convert video frames into individual images, and subsequently, the Viola-Jones algorithm was applied for region-of-interest extraction. This dataset comprises RGB images categorized into two classes: Drowsy and Non-Drowsy. Each image has dimensions of 227 x 227 pixels, resulting in a dataset that encompasses over 41,790 images.

ii) Yawning detection dataset (YAWDD):[27] The dataset includes videos recorded by an in-car camera, featuring drivers with various facial characteristics (male and female, with and without glasses/sunglasses, different ethnicities) in various situations. The videos are split into two sets. In the first set (322 videos), the camera is under the front mirror, capturing different driving scenarios: 1) normal driving (no talking), 2) talking or singing while driving, and 3) yawning while driving. Subjects contribute 3 or 4 videos each. The second set (29 videos) has the camera on the driver's dashboard, with one video per subject. Each video in this set shows driving silently, driving while talking, and driving while yawning.

2) *Collected open-source image database:* Figure 19

This dataset has been sourced from various open-source image databases to ensure inclusivity across different demographic categories, addressing identified gaps in features such as black features, narrow eyes, individuals wearing hijab, and those with glasses. The collected images were added to supplement the dataset, resulting in a total of 650 images. This comprehensive collection aims to provide a varied and representative data set of images to facilitate model training and validation across various characteristics.

3) *Students collected image dataset:* Figure 20

It is an Egyptian dataset which was gathered due to the insufficiency of existing datasets for college students and the necessity for localized Egyptian data, aiming to enhance the model's accuracy during training and validation processes. This dataset consists of 81 cases, 60 males and 21 females, with total 254 images. This dataset comprises images of undergraduate Egyptian college students at the Faculty of Engineering, Cairo University, spanning ages from 18 to 25 years old. Various categories were considered, including individuals with black features, narrow eyes, those wearing hijab, and individuals with glasses. To ensure ethical considerations, a survey was administered, seeking students' permission to use their samples for academic research. Upon receiving consent, we contacted the participants, presented them with images depicting different cases (awake, sleep, yawn), and requested them to share similar images. In cases where this wasn't feasible, we offered assistance in capturing the images by arranging meetings at the college. For data availability, the dataset generated during the current study is not publicly available due to subjects' privacy but is available from the authors on reasonable request.

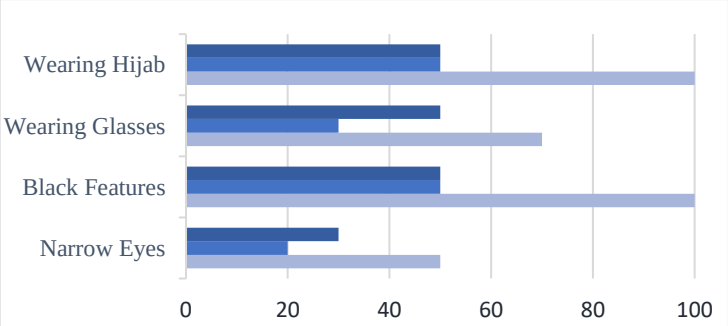
B. *Facial Landmarks Model:*

Facial landmarks are special points located by computer vision techniques to detect faces expressions and facial features tracking. The model used in this project is shape predictor 68 face landmarks which uses 68 landmarks. This model was imported from Dlib library which is designed for developing software that deals with image processing, computer vision, machine learning, and deep learning[28]. Landmarks can detect eyes, eyebrows, noses, mouth and jawlines[28]. Distances between these points are then calculated to analyze faces effectively.

1) Data preprocessing:

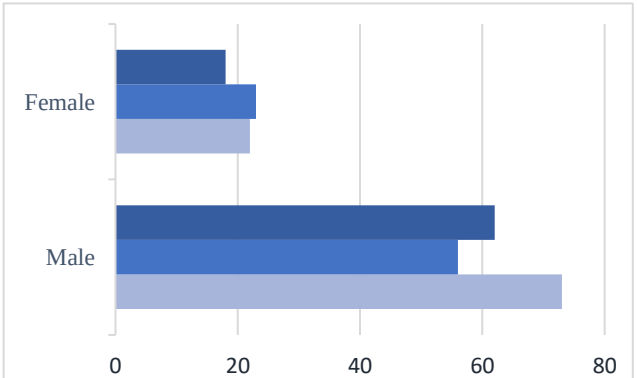
Preprocessing images plays a crucial role in the quality of results, as it ensures data consistency, enhances quality, and optimizes model training.[29] The steps of the preprocessing are listed below:

- For videos, frames were extracted at specific rates, as the neural network input cannot be videos.
- Faces were extracted from the images using Multitask Cascading Convolutional Neural Networks (MTCNN), a deep learning-based algorithm specifically designed for face detection.[30] This filtration enhances model accuracy, reduces noise, and ensures privacy while efficiently capturing essential features for emotion detection.
- The cropped faces, in the form of arrays, are converted into images using PIL, an open-source library for image processing tasks[31].



	Narrow Eyes	Black Features	Wearing Glasses	Wearing Hijab
yawn	30	50	50	50
Sleep	20	50	30	50
Active	50	100	70	100

Figure 19 Open-Source Collected images.



	Male	Female
Active	62	18
Sleep	56	23
Yawn	73	22

Figure 20 Acquired Dataset

- Images are resized to be 224 × 224 pixels in dimensions using the OpenCV library. This resizing step is important to align with the input specification of the network.
- Normalization of each pixel is done using the mean and standard deviation values of the inputs. This ensures stable training, better generalization, and efficient optimization for neural networks.[32]

$$pixel_{normalized} = \frac{(pixel_{original} - mean_{channel})}{\sigma_{channel}} \quad (17)$$

- Subsequently, mean subtraction was applied, involving the subtraction of the mean values for each channel (red, green, blue) from their respective pixel values. This centers the data around zero and aids in mitigating the influence of brightness variations.
- Images are then converted to grayscale using OpenCV. Subsequently, they are passed to the MMOD human face detector—a pretrained deep-learning-based model utilizing CNN for face detection tasks. [33]
- To locate the 68 face landmarks, a shape predictor is used. This model is imported from the Dlib library and can determine the coordinates of each landmark with respect to the image size.[34]
- Finally, the Euclidean distances between each landmark and the others are computed and saved in the form of an array with 68*68 features.

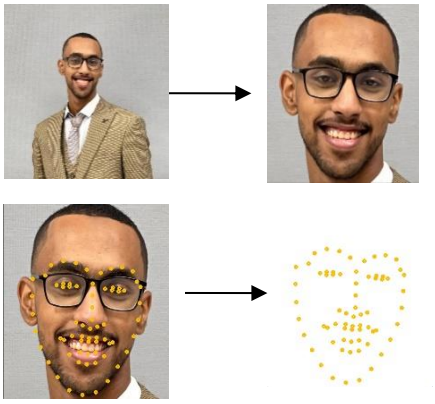


Figure 21 Face Landmarks extraction

2) ***Model Archticture:***

The neural network model was suitable for classification. This model consists of 4 hidden Dense layers of 512 neurons, all of ReLU activation function, a dropout of rate 0.4, and an output Softmax layer of 3 neurons. [Table 1] As mentioned in CNN architecture, sparse categorical cross-entropy loss was used as a loss type, with a 0.0001 learning rate.

Layer (type)	Output shape	Number of parameters
dense (Dense|)	512	1166848
dense_1 (Dense|)	512	262656
dense_2 (Dense|)	512	262656
dense_3 (Dense|)	512	262656
dropout(Dropout)	512	0
dense_4 (Dense)	3	1539

Table 1 : Facial landmarks model summary

Total number of parameters	1956355
Number of trainable parameters	1956355
Number of non-trainable parameters	0

Table 2: Number of parameters

C. VGG-16 Transfer Learning Model:

Earlier, it was explained that using pre-trained models to assist new models in specific tasks yields efficient results. VGG16, a pretrained CNN model, has been employed in various previous studies for detecting student disengagement, achieving commendable results.

1) Data preprocessing:

The initial steps of preprocessing the input images mirror those of the landmarks-based model. After normalization, pixel channels were reordered from RGB to BGR. This adjustment was made to align with the expected input format of VGG16. Subsequently, pixel values were scaled by a factor of 1/255.0 to ensure that the model's weights fall within a reasonable range. Using data loader was imperative to prevent model crashes, as the data was uploaded in separate batches.

2) Fine Tunnig:

Initially, the top layers of the VGG16 model were removed. Subsequently, a Flatten layer was added to transform the multidimensional output from the preceding convolutional and pooling layers into a one-dimensional vector [35]. Following this, two dense layers, each comprising 512 neurons and utilizing ReLU activation, were introduced. To mitigate overfitting, a dropout layer with a rate of 0.4 was incorporated [36]. Finally, a Softmax layer with three neurons was appended as the output layer. [Table 3] The model utilized the Adam optimization function, and the loss function employed was sparse categorical crossentropy, given the integer-labeled categories requiring multiclassification.

Layer (type)	Output shape	Num. of parameters
Vgg16 (Functional)	512	14714688
Flatten (Flatten)	512	0
dense (Dense|)	512	262656
dense_1 (Dense|)	512	262656
dropout (Dropout)	512	0
dense_2 (Dense)	3	1539

Table 3:Fine-tuned model

Total number of parameters	15241539
Number of trainable parameters	526851
Number of non-trainable parameters	14714688

Table 4: number of parameters

VI. Results And Analysis

Both proposed models were trained on the same dataset and in the same training environment to ensure accurate comparison of results. The performance of each model was analyzed individually before a comprehensive comparison was made between them.

In model evaluation, we relied on standard statistical methods for results analysis, including accuracy, precision, recall, and F-score for testing our models.[Table 5] These values were calculated based on a confusion matrix, described as "a square matrix that reports the counts of the true positive (TP), true negative (TN), false positive (FP), and false negative (FN) predictions of a classifier".[37]

Given that our models were performing a multiclassification task, the scoring metrics were extended to fit the multi-class problem via One Vs All (OvA) classification. Due to class imbalances where labels vary in instances, macro averaging was chosen as a more suitable approach.[37]

POC	Equation	Classification Type
Accuracy (ACC)	$\frac{TP+TN}{FP+FN+TP+TN}$	Binary
Error (ERR)	$\frac{FP+FN}{FP+FN+TP+TN}$	Binary
Precision (PRE)	$\frac{TP}{TP+FP}$	Binary
Recall	$\frac{TP}{FN+TP}$	Binary
F1 Score	$2\frac{PRE \times REC}{PRE+REC}$	Binary
Micro precision	$\frac{TP_1+\cdots+TP_k}{TP_1+\cdots+TP_k+FP_1+\cdots+FP_k}$	Multi-Class
Macro precision	$\frac{PRE_1+\cdots+PRE_k}{k}$	Multi-Class

Table 5 Model Performance Equations

The Vgg16 transfer learning model was trained on different batch sizes, but the 16-patch size achieved best results and to reach the optimum number of epochs early stop algorithm was used and stopped at the epoch number 30. So, the 16-batch size and 30 epochs performed better than the other settings. 5% of the data were used for testing and the remaining data were used for training (85%) and validation (5%). It achieved total accuracy of 93.75%, F1-Score of 93.47%, And performed well in detecting the three classes with best performance in Class 2 detection with accuracy of 96.88%

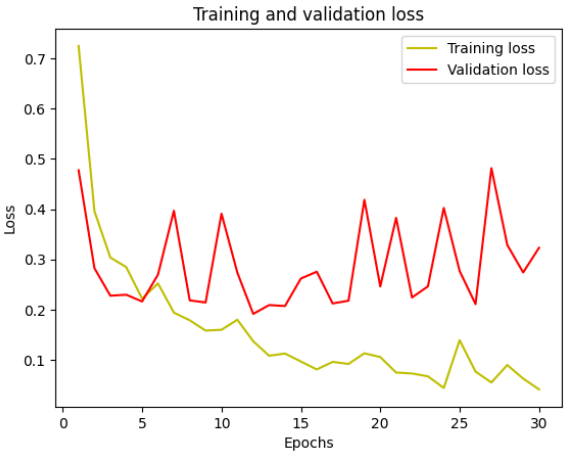


Figure 22 VGG16 Transfer Learning Model Training Vs Validation loss

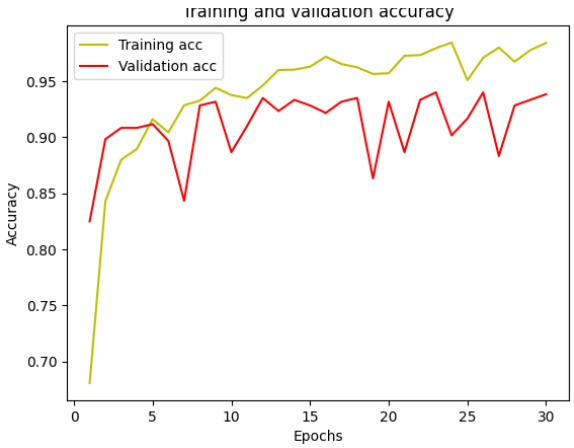


Figure 23 VGG16 Transfer Learning Model Vs Training Accuracy

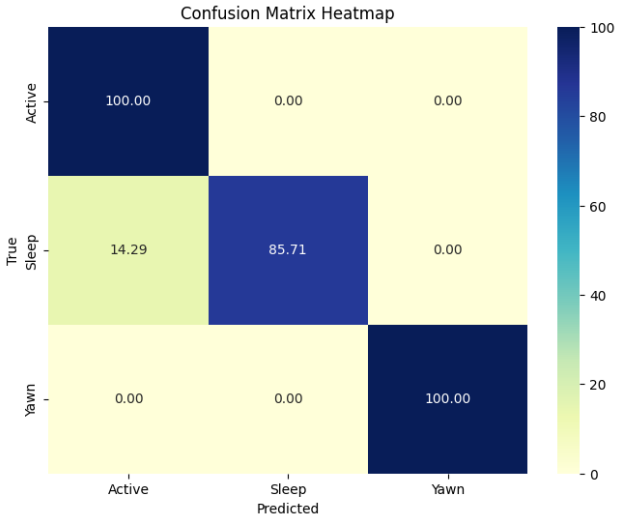


Figure 24 VGG16 Transfer Learning Model Heatmap

The Landmarks-Model was trained in patch size 64 and various number of epochs (50 – 100 – 200 – 300 – 400) as the number of epochs increases the model suffers from overfitting the 300 epochs with 0.4 drop-out rate achieved the best performance achieving 89.77% average accuracy, 85.48% F1-

Score and best performance at detecting yawing cases (class 2) with accuracy of 96.02%, 0.925 precision and 0.902 recall.

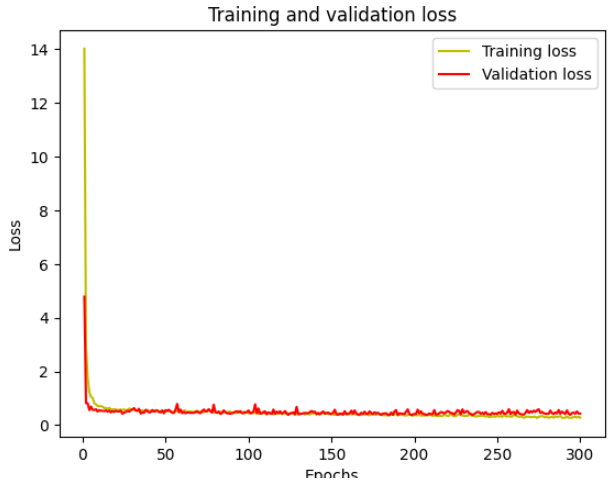


Figure 25 Facial landmarks-based Model Training Vs Validation loss

Figure 27 Facial landmarks-based Model Training Vs Validation Accuracy

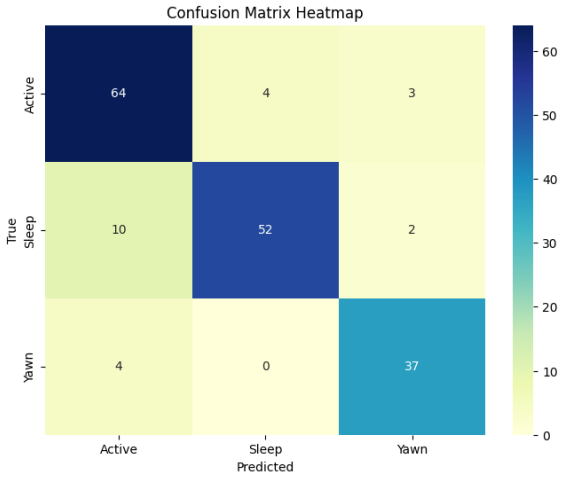


Figure 26 Facial Landmarks based Model Heatmap

After comparing both models results [Table 6] [Table 7] it's clear that the VGG16 transfer learning model achieves better results and better accuracy with less loss and less overfitting. It has achieved good performance at detecting the three levels of engagement which is clear that it is preferred to be used in the proposed system for engagement detection.

POC	VGG16	Landmarks
Accuracy	93.75%	84.66%
F1 Score	93.47%	85.48%
Class 0 accuracy	96.88%	86.36%
Class 1 accuracy	93.75%	86.93%
Class 2 accuracy	96.88%	96.02%
Average accuracy	95.83%	89.77%
Micro precision	93.75%	89.2%
Macro precision	95.56%	90.0%

Table 6 Comparison between the two models

POC	Classes	VGG16	Landmarks
Precision	Class 0	1.00	0.805
	Class 1	0.90	0.847
	Class 2	0.889	0.925
Recall	Class 0	0.928	0.873
	Class 1	0.90	0.781
	Class 2	1.00	0.902

Table 7 Precision and Recall of the two models

VII. PROPOSED SYTEM

As stated in our literature review, we seek to design a system for real-time disengagement detection in online learning setting. Our system is supposed to detect drowsiness, yawing and active states of the students as the two first classes contrast the disengagement. If the student is totally out of the frame and the model cannot detect his face it is labeled as absent.

The system is proposed in the form of web application which was developed using flask framework and JavaScript then it's connected with a user interface designed by HTML and CSS.

The system performs two main functions, Real-time drowsiness detection via webcam, Eye state analysis. Firstly, it identifies whether there is a person inside the frame or not by face detection using a Haar cascade classifier. If it couldn't detect any face in the frame, it starts to try to analyze the state of eyes by using another Haar cascade classifier for eye detection. If the aspect ratio of the eye falls below a certain threshold, it's considered closed, and the person identified as "sleep eye". If no eyes are detected the person labeled as "absent".

For each detected face, it:

- Identifies eyes using another Haar cascade classifier.
- Extracts the cropped face image.
- Resizes the image to 224x224.
- Converts the image to a NumPy array.
- Feeds the image to the pre-trained VGG16 model for disengagement detection.

- Analyzes the prediction to determine the state (active, yawn, sleepy, etc.). [Figure 28]
- Displays The detected state on the screen and sends it back to the front-end via Flask.

All the engagement and status data is saved into a data structure for each session and can be extracted and analyzed for further understanding of students' behaviors or assessment. The code of the system is saved.

Figure 28 The proposed system of detection in real-time

VIII. Conclusion

In this study, we aimed to develop an automated system for the real-time assessment of student engagement during online classes. We observed a gap in existing research regarding the detection of students' engagement levels based on their drowsiness status. Our system is supposed to address this gap using deep learning models by analyzing facial expressions and drowsiness indicators. Specifically, we focused on comparing two widely used appearance-based models: VGG16 transfer learning model and the Facial Landmarks based neural network. To ensure the effectiveness of our models, we conducted training on diverse datasets. To address inclusivity gaps in existing datasets, we carefully collected images, specifically focusing on underrepresented features like black facial characteristics, narrow eyes, hijab wearers, and individuals with glasses. Additionally, Due to the insufficiency of existing datasets for college students and the necessity for localized Egyptian data, we collected an Egyptian dataset comprising images of undergraduate students at the Faculty of Engineering, Cairo University.

Both proposed models underwent training on identical datasets and environments for accurate comparison. The Landmarks-Model exhibited notable performance, achieving an average accuracy of 89.77%, an F1-Score of 85.48%, and excelling in detecting yawning cases with 96.02% accuracy, 0.925 precision, and 0.902 recall. On the other hand, the VGG16 transfer learning model showed superior results, achieving a total accuracy of 93.75%, an F1-Score of 93.47%, and notably powerful performance in detecting different engagement levels, especially Yawning cases, with 96.88% accuracy. Comparison between both models emphasized the VGG16 transfer learning model's superiority due to its better accuracy, reduced loss, and minimal overfitting. Consequently, the VGG16 model was used in our real-time disengagement detection web application. This application classifies students into two main categories: engaged and disengaged. Disengaged students are further classified into three states: yawning, drowsy and absent, while engaged students have a single state which is active.

IX. Future Work

For future work, our primary focus is on two aspects: enhancing DL models performance and introducing and implementing the web app solution for online conferences that assess engagement.

Enhancing Facial Landmarks based model can be achieved by training the Landmarks data on different ML models and DL architectures. Also, acquiring more local data, specifically targeting students, will ensure our model is well trained in the local context. This includes collecting additional data relevant to the demographic and cultural characteristics of the student population. Moreover, training the model on various disengagement behaviors beyond drowsiness and yawning. This may include facial expressions indicating boredom or confusion, lack of eye contact with the screen, frequent distractions, minimal participation in discussions.

We aim to develop the website into a comprehensive video conference platform, aiming to create a dynamic and interactive environment that facilitates effective communication between hosts (instructors or presenters) and students. This platform will be specially designed to improve the learning experience and evaluate student engagement. It will also have distinctive features like: Real-time Engagement Assessment, Post-Session Feedback and Analytics of students' engagement.

References

[1] "UNSDG | Policy Brief: Education during COVID-19 and beyond." Accessed: Oct. 20, 2023. [Online]. Available: https://unsdg.un.org/resources/policy-brief-education-during-covid-19-and-beyond

[2] W. Leal Filho *et al.*, "COVID-19: the impact of a global crisis on sustainable development teaching," *Environ. Dev. Sustain.*, vol. 23, no. 8, pp. 11257–11278, Aug. 2021, doi: 10.1007/s10668-020-01107-z.

[3] "Student Enrollment - What is the percent of students enrolled in distance education courses in postsecondary institutions in the fall?" Accessed: Oct. 27, 2023. [Online]. Available: https://nces.ed.gov/ipeds/TrendGenerator/app/build-table/2/42?rid=87&cid=85

[4] U. Das, "Online Learning: Challenges and Solutions for Learners and Teachers," *Manag. Labour Stud.*, vol. 48, no. 2, pp. 210–213, May 2023, doi: 10.1177/0258042X211069501.

[5] M. Alawamleh, L. Al-Twait, and G. Al-Saht, "The effect of online learning on communication between instructors and students during Covid-19 pandemic," *Asian Educ. Dev. Stud.*, vol. ahead-of-print, Aug. 2020, doi: 10.1108/AEDS-06-2020-0131.

[6] C. Abla and B. R. Fraumeni, "Student Engagement: Evidence-Based Strategies to Boost Academic and Social-Emotional Results," McREL International, Nov. 2019. Accessed: Oct. 20, 2023. [Online]. Available: https://eric.ed.gov/?id=ED600576

[7] F. M. Newmann, "Reducing Student Alienation in High Schools: Implications of Theory," *Harv. Educ. Rev.*, vol. 51, no. 4, pp. 546–64, Nov. 1981.

[8] N. Bosch, "Detecting Student Engagement: Human Versus Machine," *Proc. 2016 Conf. User Model. Adapt. Pers.*, pp. 317–320, Jul. 2016, doi: 10.1145/2930238.2930371.

[9] R. D. Axelson and A. Flick, "Defining Student Engagement," *Change Mag. High. Learn.*, vol. 43, no. 1, pp. 38–43, Dec. 2010, doi: 10.1080/00091383.2011.533096.

[10] J. D. Finn and K. S. Zimmer, "Student Engagement: What Is It? Why Does It Matter?," in *Handbook of Research on Student Engagement*, S. L. Christenson, A. L. Reschly, and C. Wylie, Eds., Boston, MA: Springer US, 2012, pp. 97–131. doi: 10.1007/978-1-4614-2018-7_5.

[11] J. A. Fredricks, P. C. Blumenfeld, and A. H. Paris, "School Engagement: Potential of the Concept, State of the Evidence," *Rev. Educ. Res.*, vol. 74, no. 1, pp. 59–109, 2004.

[12] A. R. Anderson, S. L. Christenson, M. F. Sinclair, and C. A. Lehr, "Check & Connect: The importance of relationships for promoting engagement with school," *J. Sch. Psychol.*, vol. 42, no. 2, pp. 95–113, Mar. 2004, doi: 10.1016/j.jsp.2004.01.002.

[13] M. A. A. Dewan, M. Murshed, and F. Lin, "Engagement detection in online learning: a review," *Smart Learn. Environ.*, vol. 6, no. 1, p. 1, Jan. 2019, doi: 10.1186/s40561-018-0080-z.

[14] J. Yang and J. P. Gyekis, "COOPER, H.M. (2009). Research Synthesis and Meta-analysis: A Step-by-Step Approach (Applied Social Research Methods).," *Psychometrika*, vol. 77, no. 4, pp. 849–850, Oct. 2012, doi: 10.1007/s11336-012-9267-3.

[15] P. Kim, "Convolutional Neural Network," in *MATLAB Deep Learning: With Machine Learning, Neural Networks and Artificial Intelligence*, P. Kim, Ed., Berkeley, CA: Apress, 2017, pp. 121–147. doi: 10.1007/978-1-4842-2845-6_6.

[16] R. Shyam, "Convolutional Neural Network and its Architectures," vol. 12, p. 2021, Oct. 2021, doi: 10.37591/JoCTA.

[17] H. Zhang, A. Jolfaei, and M. Alazab, "A Face Emotion Recognition Method Using Convolutional Neural Network and Image Edge Computing," *IEEE Access*, vol. 7, pp. 159081–159089, 2019, doi: 10.1109/ACCESS.2019.2949741.

[18] "7.3. Padding and Stride — Dive into Deep Learning 1.0.3 documentation." Accessed: Nov. 26, 2023. [Online]. Available: http://d2l.ai/chapter_convolutional-neural-networks/padding-and-strides.html

[19] "5.3. Forward Propagation, Backward Propagation, and Computational Graphs — Dive into Deep Learning 1.0.3 documentation." Accessed: Nov. 16, 2023. [Online]. Available: http://preview.d2l.ai/d2l-en/master/chapter_multilayer-perceptrons/backprop.html

[20] "Deep Learning." Accessed: Nov. 19, 2023. [Online]. Available: https://www.deeplearningbook.org/

[21] "The Math behind Neural Networks - Forward Propagation | Jason {osa-jima}." Accessed: Nov. 19, 2023. [Online]. Available: https://www.jasonosajima.com/forwardprop

[22] "ReLU — PyTorch 2.1 documentation." Accessed: Nov. 19, 2023. [Online]. Available: https://pytorch.org/docs/stable/generated/torch.nn.ReLU.html

[23] B. Cook, "How to use the PyTorch sigmoid operation," Sparrow Computing. Accessed: Nov. 19, 2023. [Online]. Available: https://sparrow.dev/pytorch-sigmoid/

[24] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition." arXiv, Apr. 10, 2015. doi: 10.48550/arXiv.1409.1556.

[25] A. Hosna, E. Merry, J. Gyalmo, Z. Alom, Z. Aung, and M. A. Azim, "Transfer learning: a friendly introduction," *J. Big Data*, vol. 9, no. 1, p. 102, Oct. 2022, doi: 10.1186/s40537-022-00652-w.

[26] I. Nasri, M. Karrouchi, H. Snoussi, K. Kassmi, and A. Messaoudi, "Detection and Prediction of Driver Drowsiness for the Prevention of Road Accidents Using Deep Neural Networks Techniques," in *WITS 2020*, S. Bennani, Y. Lakhrissi, G. Khaissidi, A. Mansouri, and Y. Khamlichi, Eds., in Lecture Notes in Electrical Engineering. Singapore: Springer, 2022, pp. 57–64. doi: 10.1007/978-981-33-6893-4_6.

[27] M. Omidyeganeh *et al.*, "Yawning Detection Using Embedded Smart Cameras," *IEEE Trans. Instrum. Meas.*, vol. 65, no. 3, pp. 570–582, Mar. 2016, doi: 10.1109/TIM.2015.2507378.

[28] "Laskar and Sarma - 2019 - Facial Landmark Detection for Expression Analysis.pdf." Accessed: Dec. 08, 2023. [Online]. Available:

https://www.ijcseonline.org/pub_paper/272-IJCSE-07275.pdf

[29] W. Bieniecki, S. Grabowski, and W. Rozenberg, "Image Preprocessing for Improving OCR Accuracy," in *2007 International Conference on Perspective Technologies and Methods in MEMS Design*, May 2007, pp. 75–80. doi: 10.1109/MEMSTECH.2007.4283429.

[30] K. Zhang, Z. Zhang, Z. Li, and Y. Qiao, "Joint Face Detection and Alignment using Multi-task Cascaded Convolutional Networks," *IEEE Signal Process. Lett.*, vol. 23, no. 10, pp. 1499–1503, Oct. 2016, doi: 10.1109/LSP.2016.2603342.

[31] Y. Guan, F. Zhou, and J. Zhou, "Research and Practice of Image Processing Based on Python," *J. Phys. Conf. Ser.*, vol. 1345, p. 022018, Nov. 2019, doi: 10.1088/1742-6596/1345/2/022018.

[32] S. G. K. Patro and K. K. Sahu, "Normalization: A Preprocessing Stage." arXiv, Mar. 19, 2015. doi: 10.48550/arXiv.1503.06462.

[33] D. E. King, "Max-Margin Object Detection." arXiv, Jan. 30, 2015. doi: 10.48550/arXiv.1502.00046.

[34] D. E. King, "Dlib-ml: A Machine Learning Toolkit," *J. Mach. Learn. Res.*, vol. 10, pp. 1755–1758, Dec. 2009.

[35] J. Jin, A. Dundar, and E. Culurciello, "Flattened Convolutional Neural Networks for Feedforward Acceleration." arXiv, Nov. 20, 2015. doi: 10.48550/arXiv.1412.5474.

[36] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 1929–1958, Jan. 2014.

[37] S. Raschka, *Python Machine Learning*. Packt Publishing Ltd, 2015.

X. Appendix

Our web application:

https://github.com/MohamedHisham20/Hocus_Focus