



**Ain Shams University**  
**Faculty of Computer & Information Sciences**  
**Computer Science Department**

# Facial expression analysis for online learners

**By:**

Abdullah Osman [CS]  
Hady Mosaad [CS]  
Mohammed Hany [CS]  
Khadija Essa [CS]  
Esraa Ahmed [CS]

**Under Supervision of :**

[Supervisor]  
[professor],  
Dr. Maryam Nabil Al-Berry  
Department,  
Faculty of Computer and Information Sciences,  
Ain Shams University.

[TA],  
TA.Motaz Mohammed  
Faculty of Computer and Information Sciences,  
Ain Shams University.

## **Acknowledgement**

It has been a great opportunity to gain lots of experience in real time projects, followed by the knowledge of how to actually design and analyze it. so, by working on 'Facial expression analysis for online learners' project, lots of information has been gained.

All praise and thanks to ALLAH, who provided us the ability to complete this work. We hope to accept this work from us.

We are grateful of our parents and our families who are always providing help and support throughout the whole years of study. We hope we can give that back to them.

We also offer our sincerest gratitude to our supervisors, Prof. Dr. Maryam Nabil Al-Berry and T.A. Motaz Mohammed they have supported us throughout us thesis with their patience, knowledge, and experience.

Finally, we would thank our friends and all people who gave us support and encouragement.

## Abstract

Recognizing human expressions and emotions has drawn the attention of researchers, as it indicates the manifestations of nonverbal communication. The capability of recognizing one's expressions helps in human-computer interaction, which also helps in facial expression analysis for online learners.

This can be processed by using the deep Convolutional Neural Network (CNN) approach by detecting the occurrence of facial Action Units (AUs) which represents human emotion.

The proposed model focuses on recognizing the facial expressions of **an individual from a single image** it can be used to detect facial expression in real time. And this can be happening while the learner watches an educational video.

The dataset that has been used is (**DAISEE DATASET**) which contains **9068 videos and 2723882 images**.

The system has successfully classified **four basic emotion** classes which are (engagement, frustration, confusion, and boredom). Thus, the proposed method is proven to be effective for emotion recognition.

# Table of Contents

Acknowledgement .....	1
Abstract.....	2
Chapter 1 .....	7
1- Introduction .....	7
1.1 Motivation.....	7
1.2 Problem Definition.....	8
1.3 Objective .....	9
1.4 Time Plan .....	10
1.5 Document Organization .....	10
Chapter 2 .....	11
Background .....	11
2.1 -Overview on important information.....	11
2.1.1 -Project idea.....	11
2.1.2 -Deep learning overview .....	11
2.1.3 -an overview on human brain .....	11
2.1.4 -an overview on ANN.....	12
2.1.5 -Network kinds .....	12
2.1.6 -an overview on CNN.....	13
2.2 – CNN Models used in this project .....	14
2.2.1-VGG16 model.....	14
2.2.2-NasNet model .....	15
2.2.3-Xception model .....	15
Chapter 3 .....	16
Analysis and Design .....	16
3.1 System Overview .....	16
3.1.1 System Architecture.....	16
3.1.2 Overview Architecture .....	17
3.1.3 Functional Requirements:.....	18
3.1.4 Nonfunctional Requirements:.....	18
3.1.5 Preprocessing: .....	19
3.1.6 Feature Extraction. ....	19

3.1.4 Classification .....	20
3.1.7-Recognized Expression.....	21
3.2 System Analysis & Design .....	22
3.2.1 Use Case Diagram .....	22
3.2.2 Sequence Diagram .....	23
3.2.3 Class diagram .....	24
3.2.3 Database Diagram.....	25
Chapter 4 .....	26
Implementation and Testing .....	26
4.1 Preprocessing.....	26
4.1.1 Dataset split: .....	26
4.1.2 Extract Frames .....	26
4.1.3 Save file paths and labels into NumPy files. ....	27
4.2 Training .....	27
4.3 Testing.....	27
4.4 Getting the script for (new video) .....	27
4.5 Deep Learning Models.....	28
4.5.1 VGG16 .....	28
4.5.2 Xception .....	30
4.5.3 NasNet .....	32
Model Architecture: .....	33
4.6 Training .....	34
4.7 Evaluation.....	34
4.8 Results .....	36
4.9 Best Result .....	37
Chapter 5 .....	38
User Manual .....	38
5.1 steps for Running project:.....	38
Chapter 6 .....	46
Conclusion and Future Work .....	46
6.1Conclusion and Future Work: .....	46

## List of Figures

Figure 1 time plane .....	10
Figure 2 ANN one layer.....	12
Figure 3 CNN Architecture.....	13
Figure 4 how max pooling work.....	14
Figure 5 for system architecture .....	16
Figure 6 architecture for one stage.....	17
Figure 7 Convolution process .....	19
Figure 8 for classification .....	20
Figure 9 for final result .....	21
Figure 10 Use Case diagram .....	22
Figure 11 sequence diagram .....	23
Figure 12 class diagram .....	24
Figure 13 database diagram [9].....	25
Figure 14 extract frame.....	26
Figure 15 saving file path .....	27
Figure 16 model Architectur.....	28
Figure 17 model summary VGG.....	29
Figure 18 model architecture .....	30
Figure 19summary of Xception .....	31
Figure 20 nasenet Architecture .....	33
Figure 21 Accuracy.....	36
Figure 22 different way accuracy .....	37
Figure 23 splash screen.....	38
Figure 24 student or doctor .....	39
Figure 25 make new account. ....	40
Figure 26login.....	41
Figure 27 student choose video to upload.....	42
Figure 28set intervals.....	43
Figure 29 show result.....	44

## List of Abbreviations

<b>ANN : Artificial Neural Network</b> .....	11
API : Application Programming Interface .....	33
CNN : Convolution Neural Network .....	ii
ConvNet:Convolution Network .....	12
ITS: Intelligent Tutoring System .....	vii
MOOCs: Massive Open Online Courses .....	vii
<b>NasNet : stands for Neural search architicture</b> .....	13
UI : User Interface .....	9
<b>VGG : VGG stands for Visual Geometry Group</b> .....	13
<b>Xception: Extreme version of Inception</b> .....	13

# Chapter 1

## 1- Introduction

### 1.1 Motivation

Facial Expressions are the most important aspects of human communication as the Face is responsible for communicating not only thoughts or ideas but also emotions, in general, people infer the emotional states of other people, such as happiness, sadness, and anger, using facial expressions and vocal tone, according to different surveys: verbal components convey one-third of human communication, and nonverbal components convey two-thirds.

Among several nonverbal components by carrying emotional meaning.

Therefore, it is natural that research of facial emotion has been gaining lot of attention over the past decades.

The main purpose is to detect the engagement levels of online learners according to his/her facial expression which by this method it will help any education system that uses e-learning environment, it also will help the instructors to improve the educational content.

The idea is based on Convolutional Neural Network Deep Learning, which is the most applied to analyzing visual imagery, CNN nowadays can solve all the problems that can be expressed in image form, as it used in labeling someone in Facebook pictures, autonomous cars, which can “read” traffic signs, recognize other cars, and even detect if a person is crossing the street. All these functionalities are based on CNN.



## 1.2 Problem Definition

Human beings require time and persistent effort to perform a task such as learning. On the same lines, the researchers have argued the importance of continuous effort or engagement to accomplish the learning task.

In any education system, student engagement is a key component. With the advent of digital technologies, the traditional classroom teaching is transformed into advanced learning environments such as an Intelligent Tutoring System (ITS) and e-learning environment such as Massive Open Online Courses (MOOCs). As a result, Student engagement in the learning environment has also adapted in its own way.

Student engagement is defined as a complex structure with multi-dimensions and components. Various other works have classified it in separate ways.

Several traditional methods and measures introduced in literature to assess the level of

Engagement, have their own relevance. Different methods such as self-reports: participants answer the set of questions related to their experience such as level of engagement, interest and so on. Secondly, an observational study was done by the external expert: Focus on the behavioral analysis of students.

Traditional measurement methods are not sufficient to measure engagement in all the contexts.

So, automatic engagement assessments for the digitally transformed learning environment are required.

These techniques analyze various facial cues, body posture, well known social cues of engagement and disengagement captured automatically using affective computing techniques. These techniques are sensitive to the engagement levels variations over time.

Moreover, automatic measures can facilitate timely intervention to change the course of fading engagement level.

### **1.3 Objective**

The main objectives of our project are summarized as

- Building a System that detects human's facial expression by extracting the features from face to detect whether it belongs to (engagement, frustration, confusion, and boredom).
- This will provide to the education systems to detect engagement levels of students to improve the performance of education system that uses e-learning environment.
- Create a system with high usability.
- Changes can be made easily to satisfy new requirements or to correct deficiencies.

## 1.4 Time Plan

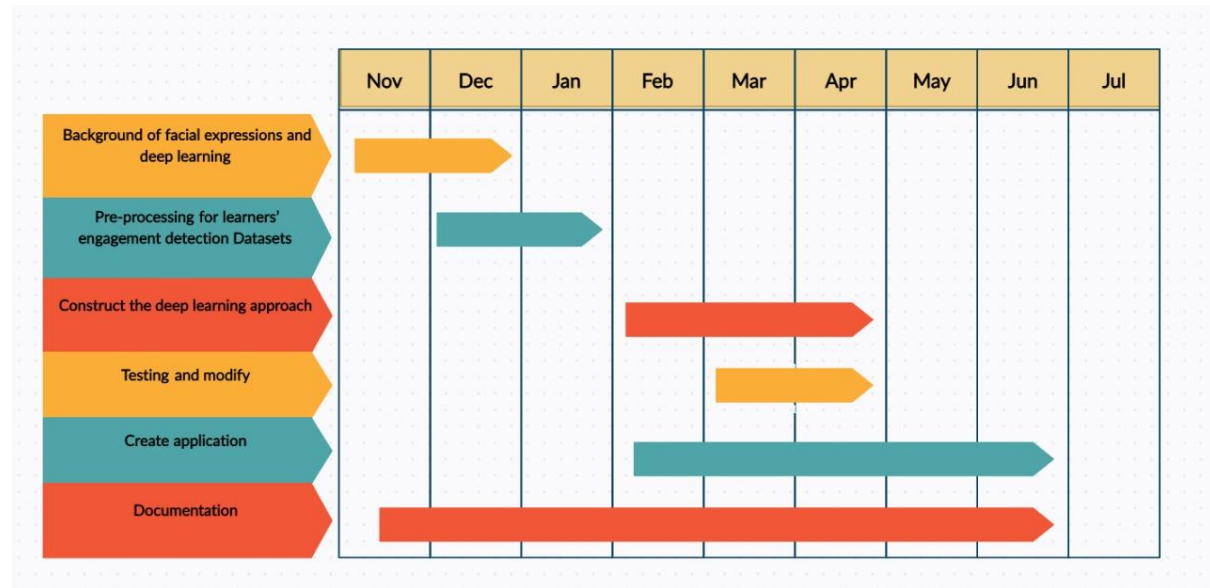


Figure 1 time plane

## 1.5 Document Organization

**This section will describe the documentation chapters.**

This documentation contains six chapters: -

**Chapter 2 includes the application background**, so it includes a detailed description of the project field, all the scientific background related to the project, related work, and a description of existing similar applications.

- **Chapter 3 includes analysis and design**, so it contains a system overview (system architecture and system users) and system analysis and design which contains system diagrams (use case, sequence diagram).

- **Chapter 4 includes a description of all system functions implemented in the project**, a description of techniques and algorithms used UI design.

- **Chapter 5 includes the user manual** which will help the user to know how to use this application in detail and the installation guide that would describe how to install the program, and all required third-party tools that need to be available for the project to run.

- **Chapter 6 contains a conclusion and future work.**

## **Chapter 2**

### **Background**

#### **2.1 -Overview on important information**

##### **2.1.1 -Project idea**

Project idea is based on Deep learning which is an artificial intelligence function that simulates the working of the human brain in processing data.

##### **2.1.2 -Deep learning overview**

Deep learning approaches are a class of machine learning algorithms that use many layers of nonlinear processing units for representations and transformations. Each layer uses the output from the previous layer as input since Deep word means number of hidden layers in network.

##### **2.1.3 -an overview on human brain**

In the human brain, there are about 100 billion neurons, each neuron connects to about 100000 of its neighbors, a neuron has a body, dendrites, and an axon. The signal from one neuron travels down the axon and transfers to the dendrites of the next neuron. That connection where the signal passes is called a synapse.

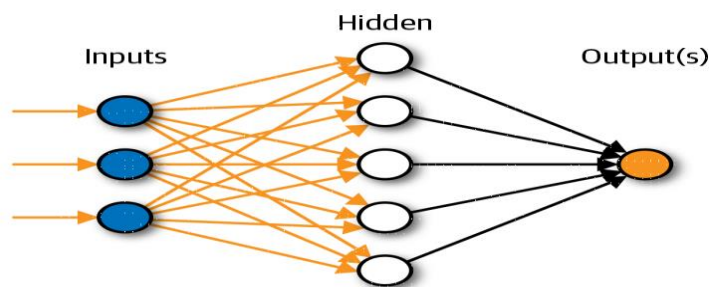
## 2.1.4 -an overview on ANN

Traditional neural networks only contain 2-3 hidden layers, while deep networks can have as many as 150.

The input node (Neuron) takes information in a numerical form.

The information is presented as an activation value where each node is given a number. [1]

**Artificial Neural Network**



*Figure 2 ANN one layer*

## 2.1.5 -Network kinds

Networks have two kinds, Feedforward and Feedback Network,

1. A feedback network has feedback paths. This means that they can have signals traveling in both directions using loops. All possible connections between neurons are allowed.
2. A feedforward network is a network that contains inputs, outputs, and hidden layers. The signals can only travel in one direction (forward). Input data passes into a layer where calculations are performed. Each processing element calculates based upon the weighted sum of its inputs. The new values become the new values that feed the next layer (feed-forward).

### 2.1.6 -an overview on CNN

A specific kind of such a deep neural network is the Convolutional Neural Network, which is commonly referred to as CNN or ConvNet, it is a feed-forward artificial neural network. CNNs specifically are inspired by the biological visual cortex. The cortex has small regions of cells that are sensitive to the specific areas of the visual field, some researchers showed that some individual neurons in the brain activated or fired only in the presence of edges of a particular orientation like vertical or horizontal edges, that's where CNN idea is comes from.

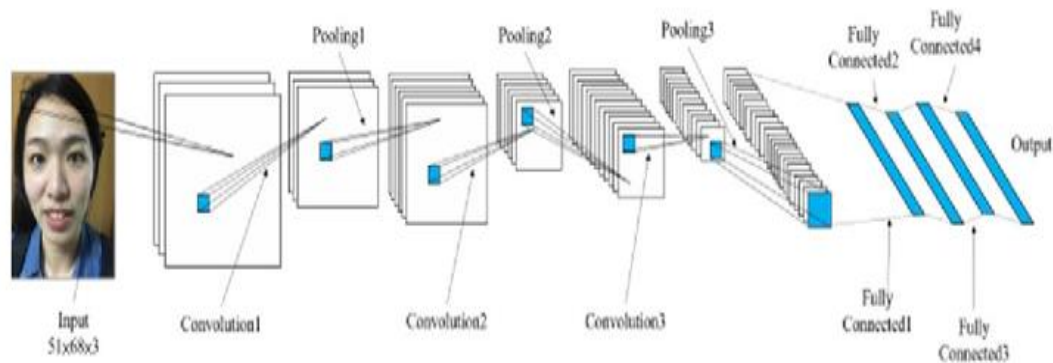


Figure 3 CNN Architecture

The objective of subsampling (i.e., Max pooling) is to get an input representation by reducing its dimensions, which helps in reducing overfitting.

The sub-sampled output of a max-pooling operation with a stride of 2 applied on an input image (I). As we know the objective of (Max pooling) is to reduce size of feature map by determine size ( $X * Y$ ) of area in a way

to pick the max value from that area that covers some area in the input image and put it in the result as we see in (figure 4) below. [2]

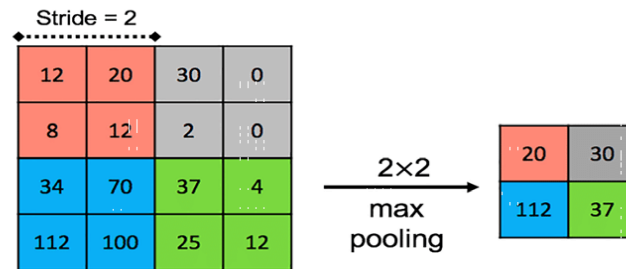


Figure 4 how max pooling work

## 2.2 – CNN Models used in this project

### 2.2.1-VGG16 model

VGG16 is a convolutional neural network model proposed by K. ... Zisserman from the University of Oxford in the paper “Very Deep Convolutional Networks for Large-Scale Image Recognition”. The model achieves 92.7% top-5 test accuracy in ImageNet, which is a dataset of over 14 million images belonging to 1000 classes. [3]

### **2.2.2-NasNet model**

Google recently launched NasNet, is currently the best model in the field of image recognition, this model has reproduced the next, but also a general understanding of its principles. This model is not artificially designed, but through Google before the introduction of automatic early AutoML trained. The project aims to achieve "automatic machine learning", i.e., training machine learning software to create software machine learning, the development of the new system on their own layer of code, it is also a neural architecture search technology (Neural Architecture Search technology). [4]

### **2.2.3-Xception model**

**Xception Model** is proposed by Francois Chollet. **Xception** is an extension of the inception Architecture which replaces the standard Inception modules with depth wise Separable Convolutions. [5]



## Chapter 3

### Analysis and Design

#### 3.1 System Overview

##### 3.1.1 System Architecture

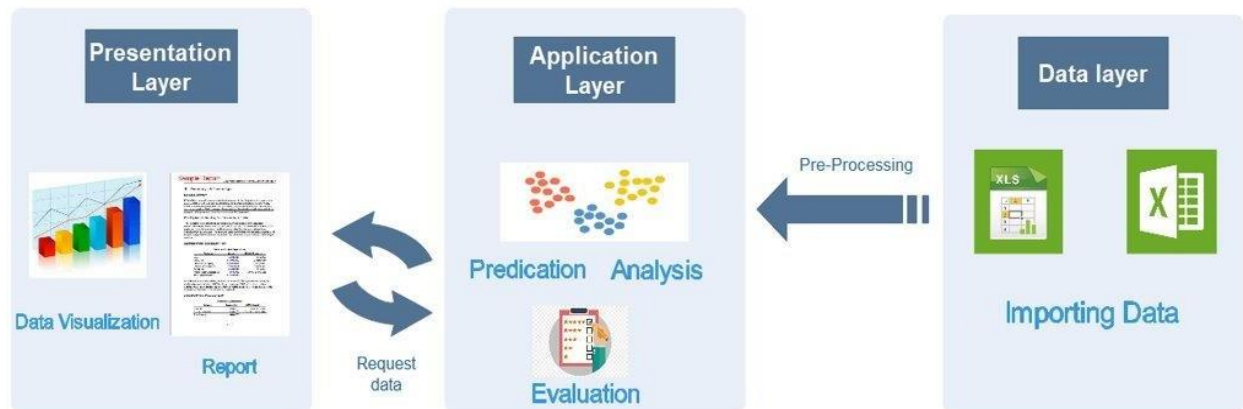


Figure 5 for system architecture

- **Presentation Layer:**

The presentation layer is the front layer in the system architecture, it consists of the user interface. This is the layer where the user interacts with an application.

- **Application Layer:**

The application layer is the layer that the presentation layer and data layer use to communicate with each other and it has all the features of the application.

- **Data Layer:**

The data layer is where the information is stored and retrieved, it can be also referred to as “Storage Tier”. The data layer is consisting of a database that stores the application data.

### 3.1.2 Overview Architecture

In General Architecture we first load our Dataset (DAiSEE dataset), then we make preprocessing on all data, preprocessing.

Then we split Dataset into training, validation, and testing data, then we need to make feature extraction before classification, but we follow Deep Learning Models that perform feature extraction and classification in the same network and do not need to make feature extraction first, because first layers in network made feature extraction and last layers made classification.

We used four deep learning models.

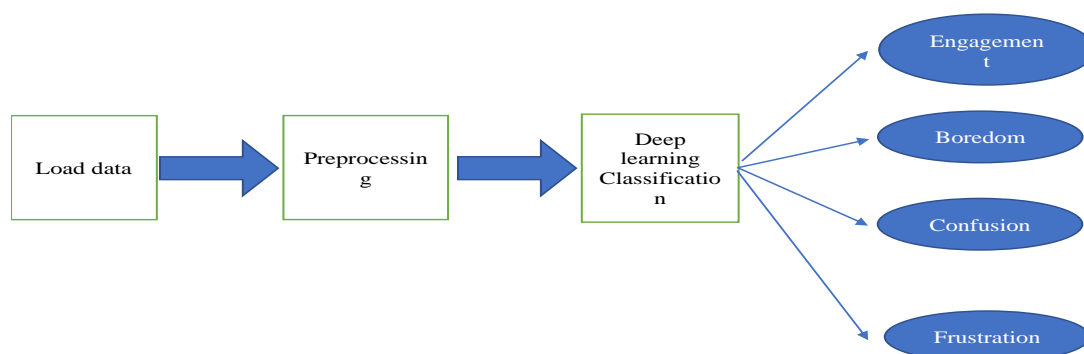
1-Xception.

2-VGG16.

3-NasNet.

Each model we tried it using three Trials of data to select best trial and best model to using it.

→ **this figure show architecture for one stage**



*Figure 6 architecture for one stage*

### **3.1.3 Functional Requirements:**

- **Recording video:**  
make record for student to get video.
- **Availably to set interval:**  
Instructor has to ability to determine a specific interval for every content in lecture.
- **Student can upload video:**  
After student watch lecture he can upload this video to server

### **3.1.4 Nonfunctional Requirements:**

- **Performance:**

The system shall respond quickly to the actions of the users and never cause any delayed response.

- **Reliability:**

The system shall handle all errors and never shut down suddenly for the users.

- **Scalability:**

The system shall be able to work efficiently even with huge amounts of data.

- **Usability:**

The system shall have an easy and clear interface to be understandable by all its users.

### 3.1.5 Preprocessing:

Depend on our dataset that interval for each video 10se we extract 7 frames. The reason for we extract only 7 frames because the emotion for human change every 1.75 second. [6]

### 3.1.6 Feature Extraction.

The main purpose of the convolution step is to extract features from the input image. The convolutional layer is always the first step in a CNN. we take the filter and apply it pixel block by pixel block to the input image. we do this through the multiplication of the matrices. There is a simple formula to do so: If dimension of an image is  $n \times n$ . If dimension of filter is  $f \times f$ . Dimension of output will be  $((n-f+1), (n-f+1))$  [7]

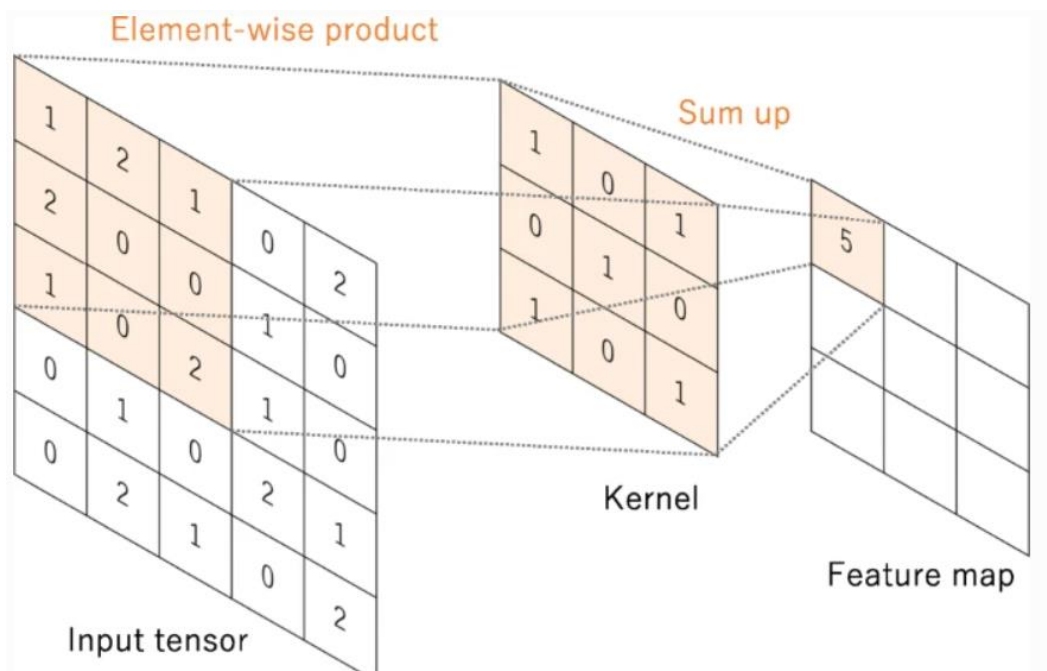


Figure 7 Convolution process

### 3.1.4 Classification

the convolution layer has extracted some valuable features from the data. These features are sent to the fully connected layer that generates the results. The output from the convolution layer was a 2D matrix. Ideally, we would want each row to represent a single input image. In fact, the fully connected layer can only work with 1D data. Hence, the values generated from the previous operation are first converted into a 1D format. Once the data is converted into a 1D array, it is sent to the fully connected layer. All these individual values are treated as separate features that represent the image. [8]

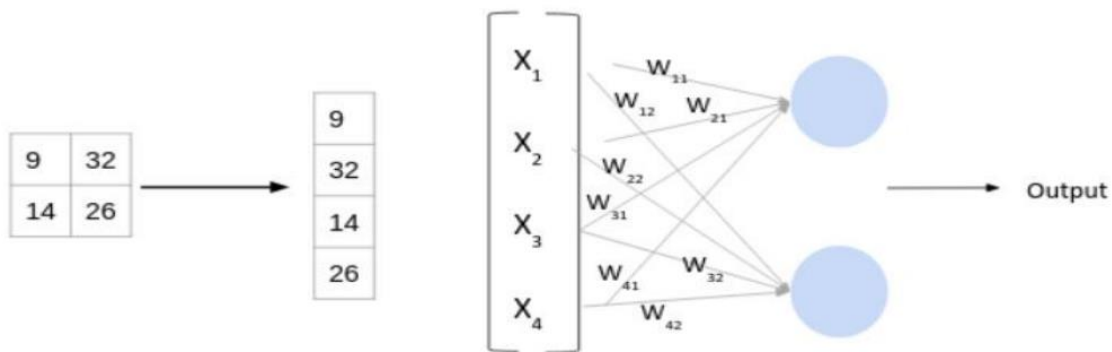


Figure 8 for classification

### 3.1.7-Recognized Expression.

The final step that determines which class the input image belongs to (Engagement, Boredom, Confusion, Frustration), by the fully connected (dense) layers that uses data from convolution layer to generate output.

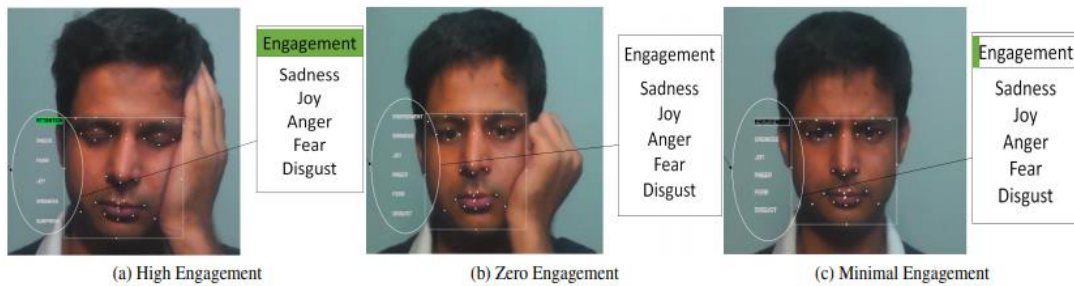
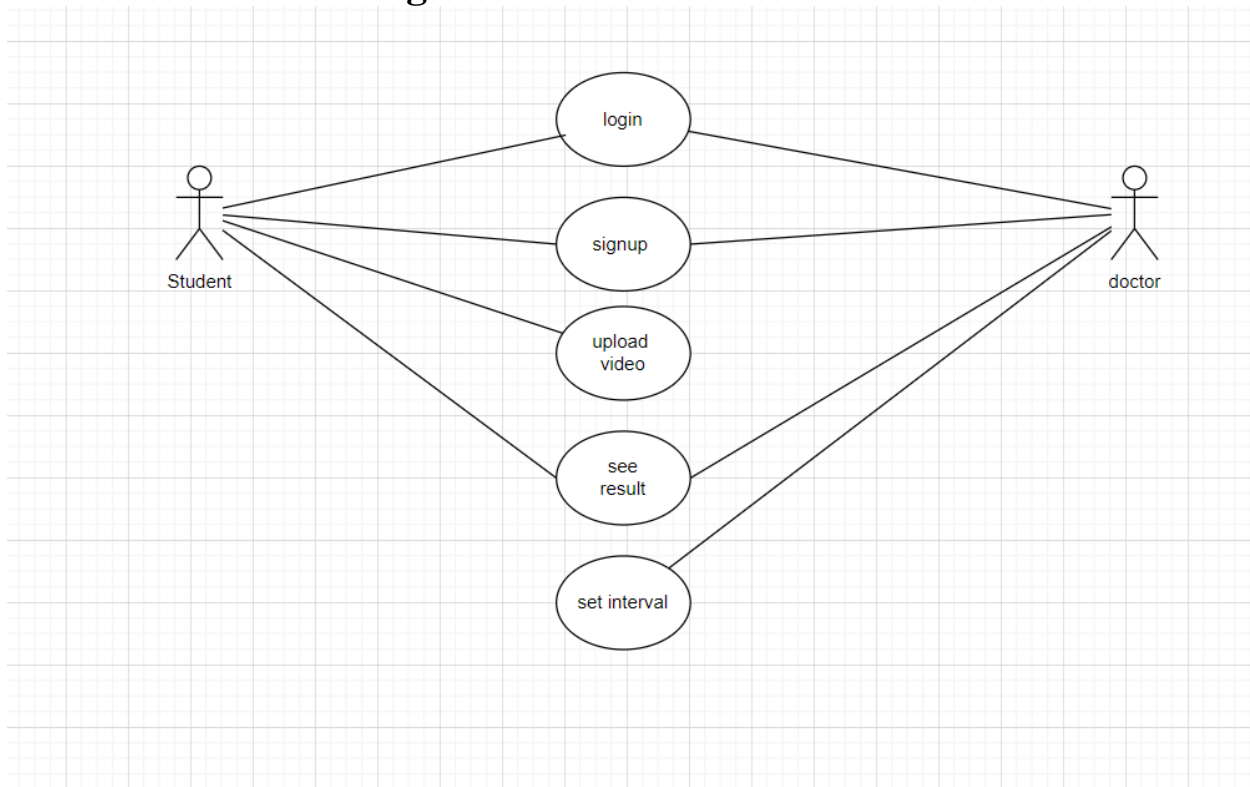


Figure 9 for final result

## 3.2 System Analysis & Design

### 3.2.1 Use Case Diagram



*Figure 10 Use Case diagram*

### 3.2.2 Sequence Diagram

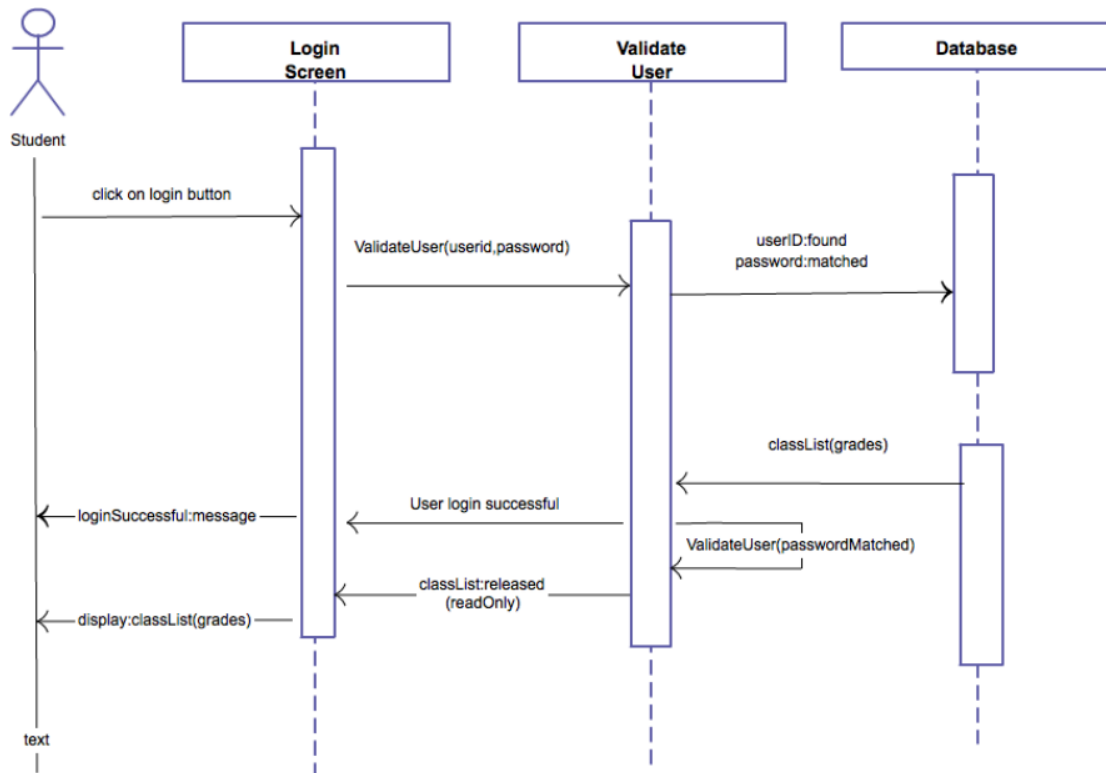


Figure 11 sequence diagram



### 3.2.3 Class diagram

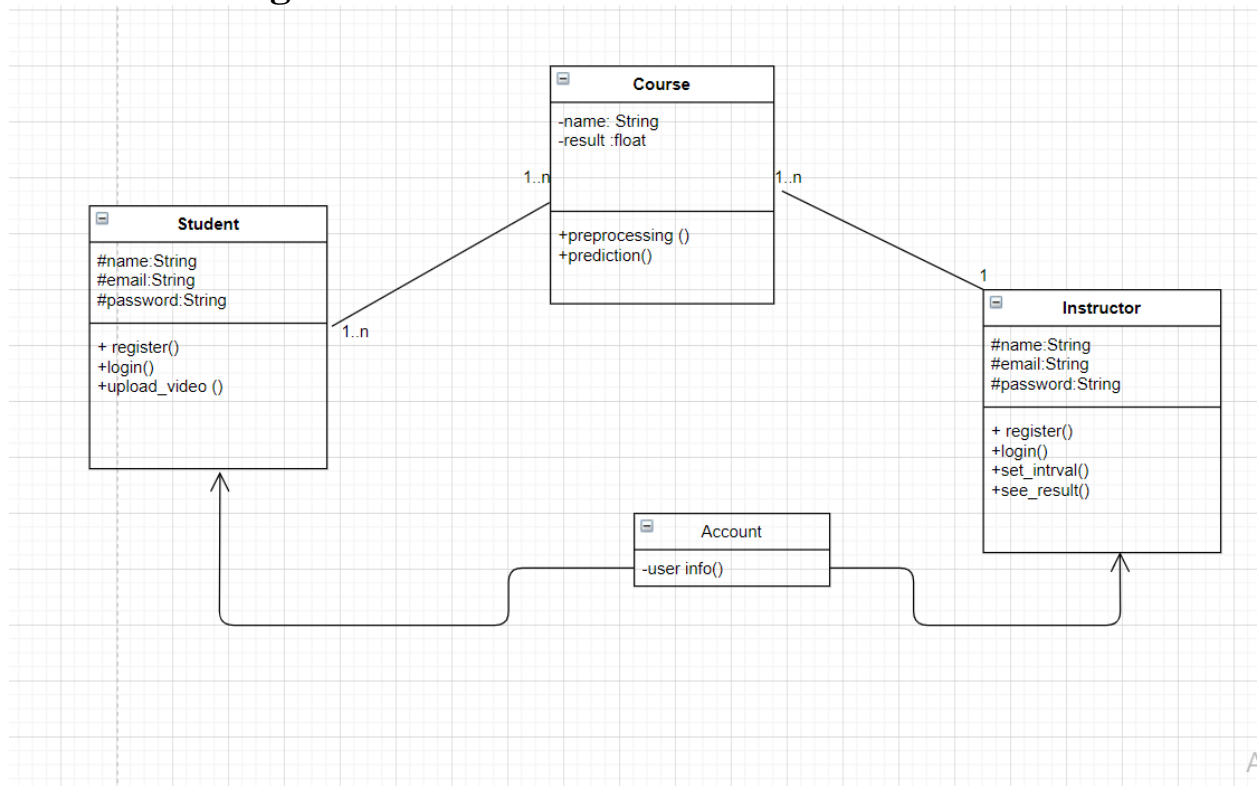


Figure 12 class diagram

### 3.2.3 Database Diagram

Using data base firebase for saving result and data

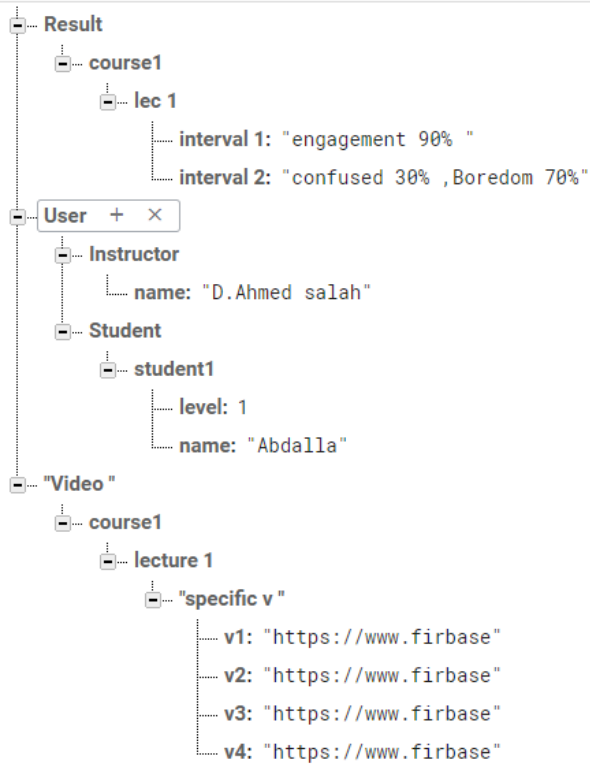


Figure 13 database diagram [9]

## Chapter 4

### Implementation and Testing

#### 4.1 Preprocessing

##### 4.1.1 Dataset split:

To prepare the dataset for the Model, we create a data split into train, validation and test sets.

**The following three principles are used:**

- For training deep learning models, follow the general Kaggle practice of 60:20:20 for the (train : validation : test )sets
- All splits are mutually exclusive and exhaustive with respect to subjects
- The same ratio of males : females is maintained across the splits

##### 4.1.2 Extract Frames

Take every video and split every 10s to 7 frames using FFmpeg which is software platform especially for multimedia files. With the help of its expanded libraries we can convert, edit, repair, format any video format.

```
Extracting frames for Train
100% (5482 of 5482) |#####| Elapsed Time: 0:10:09 Time: 0:10:09
Extracting frames for Test
100% (1866 of 1866) |#####| Elapsed Time: 0:03:18 Time: 0:03:18
Extracting frames for Validation
100% (1720 of 1720) |#####| Elapsed Time: 0:02:29 Time: 0:02:29
```

*Figure 14 extract frame*

### 4.1.3 Save file paths and labels into NumPy files.

Save file paths of all the frames and their respective output labels as NumPy array.

Namely this NumPy arrays would be x\_train, y\_train, etc.

This NumPy array would be directly used in input pipeline for training and testing of model.

```
Getting filepath and labels for Train
100% (37506 of 37506) |#####| Elapsed Time: 0:03:04 Time: 0:03:04
Getting filepath and labels for Test
100% (12488 of 12488) |#####| Elapsed Time: 0:00:28 Time: 0:00:28
Getting filepath and labels for Validation
100% (10003 of 10003) |#####| Elapsed Time: 0:00:20 Time: 0:00:20
```

Figure 15 saving file path

## 4.2 Training

- Get NumPy folder (train, validation)

- Make train to the model in (train\_ds, validation\_ds)→(Xception model, VGG16 and NasNet)

- Save the model and its weights.

## 4.3 Testing

- Get NumPy folder (test)

- Make test to model in (test\_ds)

## 4.4 Getting the script for (new video)

- Split it to interval.

- Deal with every interval as new video

- Split it into videos of 10s.

- Split every 10s to 7 frames.

- Get the result from every interval.

- Show result.

## 4.5 Deep Learning Models

We use keras deep learning Framework to include all models

**Keras** is an open-source neural-network library written in Python. It can run on top of **TensorFlow**, Microsoft Cognitive Toolkit, R, **Theano**, or **Plaid ML**. Designed to enable fast experimentation with deep neural networks, it focuses on being user-friendly, modular, and extensible. [10]

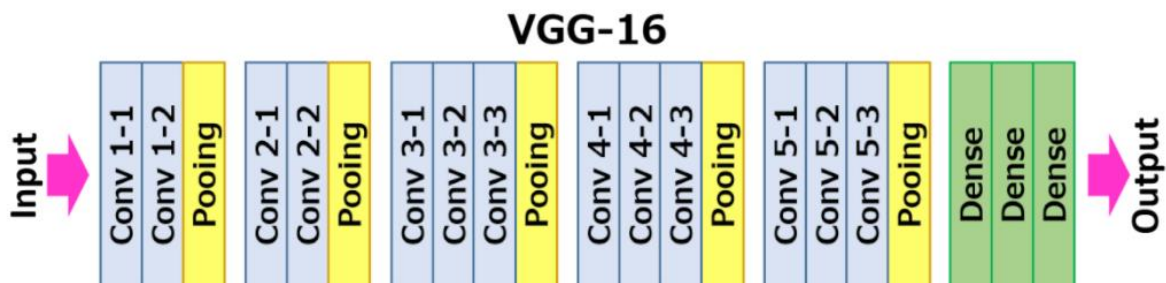
### 4.5.1 VGG16

VGG16 is a convolutional neural network model proposed by K. Simonyan and A. Zisserman from the University of Oxford in the paper “Very Deep Convolutional Networks for Large-Scale Image Recognition”.

The model achieves 92.7% top-5 test accuracy in ImageNet, which is a dataset of over 14 million images belonging to 1000 classes. It was one of the famous models submitted to ILSVRC-2014.

It makes the improvement over Alex Net by replacing large kernel-sized filters (11 and 5 in the first and second convolutional layer, respectively) with multiple 3×3 kernel-sized filters one after another.

VGG16 was trained for weeks and was using NVIDIA Titan Black GPU’s [11]



VGG16

Figure 16 model Architectur

## Model Summary

Layer (type)	Output Shape	Param #	Connected to
... input_2 (InputLayer)	[(None, 224, 224, 3)]	0	
block1_conv1 (Conv2D)	(None, 224, 224, 64)	1792	input_2[0][0]
block1_conv2 (Conv2D)	(None, 224, 224, 64)	36928	block1_conv1[0][0]
block1_pool (MaxPooling2D)	(None, 112, 112, 64)	0	block1_conv2[0][0]
block2_conv1 (Conv2D)	(None, 112, 112, 128)	73856	block1_pool[0][0]
block2_conv2 (Conv2D)	(None, 112, 112, 128)	147584	block2_conv1[0][0]
block2_pool (MaxPooling2D)	(None, 56, 56, 128)	0	block2_conv2[0][0]
block3_conv1 (Conv2D)	(None, 56, 56, 256)	295168	block2_pool[0][0]
block3_conv2 (Conv2D)	(None, 56, 56, 256)	590080	block3_conv1[0][0]
block3_conv3 (Conv2D)	(None, 56, 56, 256)	590080	block3_conv2[0][0]
block3_pool (MaxPooling2D)	(None, 28, 28, 256)	0	block3_conv3[0][0]
block4_conv1 (Conv2D)	(None, 28, 28, 512)	1180160	block3_pool[0][0]
block4_conv2 (Conv2D)	(None, 28, 28, 512)	2359008	block4_conv1[0][0]
block4_conv3 (Conv2D)	(None, 28, 28, 512)	2359008	block4_conv2[0][0]
block4_pool (MaxPooling2D)	(None, 14, 14, 512)	0	block4_conv3[0][0]
block5_conv1 (Conv2D)	(None, 14, 14, 512)	2359008	block4_pool[0][0]
block5_conv2 (Conv2D)	(None, 14, 14, 512)	2359008	block5_conv1[0][0]
block5_conv3 (Conv2D)	(None, 14, 14, 512)	2359008	block5_conv2[0][0]
block5_pool (MaxPooling2D)	(None, 7, 7, 512)	0	block5_conv3[0][0]
global_average_pooling2d_1 (GlobalAveragePooling2D)	(None, 512)	0	block5_pool[0][0]
y1 (Dense)	(None, 4)	2052	global_average_pooling2d_1[0][0]
y2 (Dense)	(None, 4)	2052	global_average_pooling2d_1[0][0]
y3 (Dense)	(None, 4)	2052	global_average_pooling2d_1[0][0]
y4 (Dense)	(None, 4)	2052	global_average_pooling2d_1[0][0]
Total params: 14,722,896			
Trainable params: 8,208			
Non-trainable params: 14,714,688			

Figure 17 model summary VGG

## 4.5.2 Xception

**Xception** is Inspired by Google's Inception model, stands for Extreme version of Inception With a modified **depth wise separable convolution**, it is **even better than Inception-v3** (also by Google, 1st Runner Up in ILSVRC 2015) for both ImageNet ILSVRC and JFT datasets.

Xception is based on an 'extreme' interpretation of the Inception model its architecture is a linear stack of depth wise separable convolution layers with residual connections Simple and modular architecture fundamental hypothesis mapping of cross-channels correlations and spatial correlations can be entirely decoupled.

composed of 36 convolutional layers forming the feature extraction base of the network structured into 14 modules, all of which have linear residual connections around them, except for the first and last modules. [12]

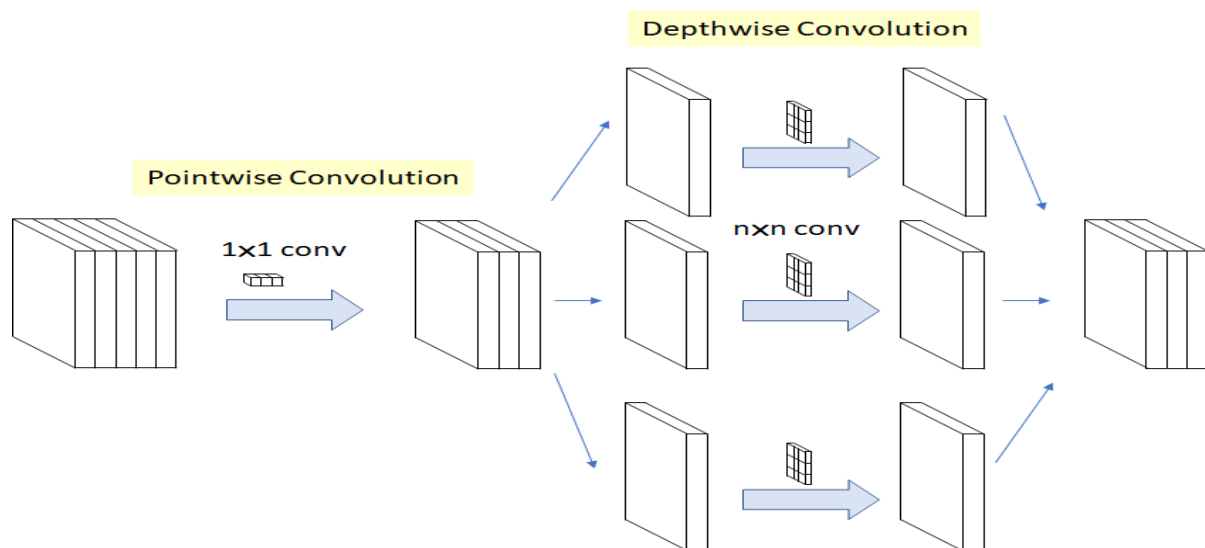


Figure 18 model architecture

**The Modified Depth Wise Separable Convolution used as an Inception Module in Xception, so called “extreme” version of Inception module (n=3 here)**



## Model Summary

add_10 (Add)	(None, 19, 19, 728)	0	block12_sepconv3_bn[0][0] add_9[0][0]
block13_sepconv1_act (Activatio	(None, 19, 19, 728)	0	add_10[0][0]
block13_sepconv1 (SeparableConv	(None, 19, 19, 728)	536536	block13_sepconv1_act[0][0]
block13_sepconv1_bn (BatchNorma	(None, 19, 19, 728)	2912	block13_sepconv1[0][0]
block13_sepconv2_act (Activatio	(None, 19, 19, 728)	0	block13_sepconv1_bn[0][0]
block13_sepconv2 (SeparableConv	(None, 19, 19, 1024)	752024	block13_sepconv2_act[0][0]
block13_sepconv2_bn (BatchNorma	(None, 19, 19, 1024)	4096	block13_sepconv2[0][0]
conv2d_3 (Conv2D)	(None, 10, 10, 1024)	745472	add_10[0][0]
block13_pool (MaxPooling2D)	(None, 10, 10, 1024)	0	block13_sepconv2_bn[0][0]
batch_normalization_3 (BatchNor	(None, 10, 10, 1024)	4096	conv2d_3[0][0]
add_11 (Add)	(None, 10, 10, 1024)	0	block13_pool[0][0] batch_normalization_3[0][0]
block14_sepconv1 (SeparableConv	(None, 10, 10, 1536)	1582080	add_11[0][0]
block14_sepconv1_bn (BatchNorma	(None, 10, 10, 1536)	6144	block14_sepconv1[0][0]
block14_sepconv1_act (Activatio	(None, 10, 10, 1536)	0	block14_sepconv1_bn[0][0]
block14_sepconv2 (SeparableConv	(None, 10, 10, 2048)	3159552	block14_sepconv1_act[0][0]
block14_sepconv2_bn (BatchNorma	(None, 10, 10, 2048)	8192	block14_sepconv2[0][0]
block14_sepconv2_act (Activatio	(None, 10, 10, 2048)	0	block14_sepconv2_bn[0][0]
global_average_pooling2d (Globa	(None, 2048)	0	block14_sepconv2_act[0][0]
y1 (Dense)	(None, 4)	8196	global_average_pooling2d[0][0]
y2 (Dense)	(None, 4)	8196	global_average_pooling2d[0][0]
y3 (Dense)	(None, 4)	8196	global_average_pooling2d[0][0]
y4 (Dense)	(None, 4)	8196	global_average_pooling2d[0][0]
=====			
Total params: 20,894,264			
Trainable params: 32,784			
Non-trainable params: 20,861,480			

Figure 19summary of Xception



### 4.5.3 NasNet

NASNet was developed by **Google Brain**, is. Authors propose to *search for an architectural building block on a small dataset* and then *transfer the block to a larger dataset*. Particularly, they search for the best convolutional layer or cell on CIFAR-10 first, then apply this cell to the ImageNet by stacking together more copies of this cell. A new regularization technique called **Scheduled Drop Path** is also proposed which significantly improves the generalization in the NASNet models. At last, **NASNet model achieves state-of-the-art results with smaller model size and lower complexity (FLOPs)** [13]

#### Model Summary

normal_add_3_18 (Add)	(None, 7, 7, 672)	0	normal_left3_18[0][0] adjust_bn_18[0][0]
normal_add_4_18 (Add)	(None, 7, 7, 672)	0	normal_left4_18[0][0] normal_right4_18[0][0]
normal_add_5_18 (Add)	(None, 7, 7, 672)	0	separable_conv_2_bn_normal_left5_ normal_bn_1_18[0][0]
normal_concat_18 (Concatenate)	(None, 7, 7, 4032)	0	adjust_bn_18[0][0] normal_add_1_18[0][0] normal_add_2_18[0][0] normal_add_3_18[0][0] normal_add_4_18[0][0] normal_add_5_18[0][0]
activation_259 (Activation)	(None, 7, 7, 4032)	0	normal_concat_18[0][0]
global_average_pooling2d (Glo	(None, 4032)	0	activation_259[0][0]
fc1 (Dense)	(None, 128)	516224	global_average_pooling2d[0][0]
fc2 (Dense)	(None, 64)	8256	fc1[0][0]
y1 (Dense)	(None, 4)	260	fc2[0][0]
y2 (Dense)	(None, 4)	260	fc2[0][0]
y3 (Dense)	(None, 4)	260	fc2[0][0]
y4 (Dense)	(None, 4)	260	fc2[0][0]
=====			
Total params: 85,442,338			
Trainable params: 525,520			
Non-trainable params: 84,916,818			

## Model Architecture:

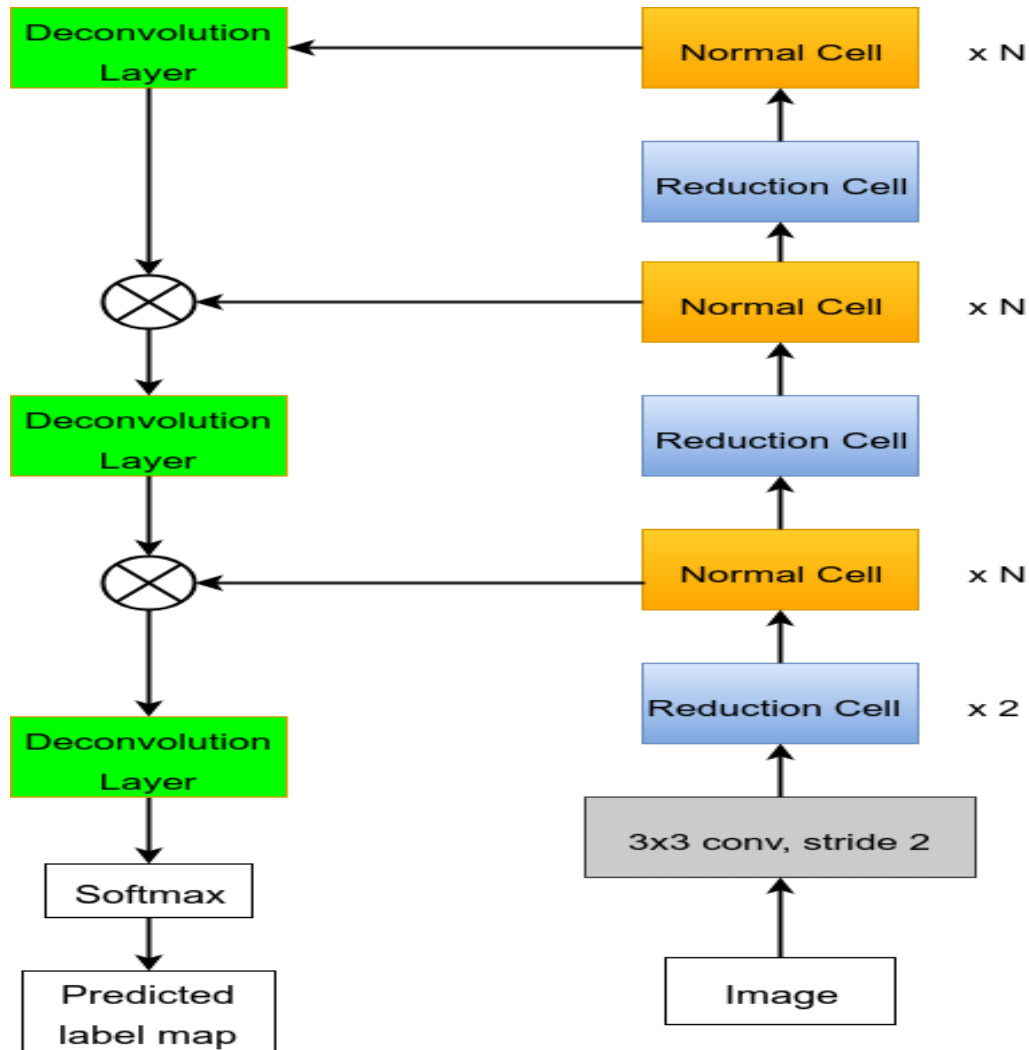


Figure 20 nasenet Architecture

## 4.6 Training

We first update the models to fit our classification problem by removing the top classification layers.

and then added our proper classification layers contains of fully connected layers with a SoftMax activation function to distribute the probability.

on 4 classes. We added this classification head in two ways,

1. Added on top of the global average pooling layer directly.
2. Added two fully connected (FC) layers between the global average pooling layer and the classification head.

## 4.7 Evaluation

We evaluate our models by calculating confusion matrix that clarify right and wrong classifications in each class.

Then we use confusion matrix to calculate overall accuracy and calculate accuracy for each class then calculate avg accuracy of all classes.

Overall Accuracy: number of correctly predicted beats/total of beats to predict.

Average Accuracy: it is the average of each accuracy per class (sum of accuracy for each class predicted/number of class)

In unbalanced datasets, overall accuracy may be not accurate to test a model.

Average accuracy is a good way to test a model to know if it has learnt all classes almost the same

### **i. Build Confusion Matrix**

- Given actual classes and predictions output from model
- Define 2D array of zeros of size classes×classes
- Iterate on actual and predictions
- Row index = actual[I]
- Column index = predictions[I]
- Increment confusion\_matrix [Row index, Column index]

### **ii. Calculate Accuracy**

- Given Confusion Matrix
- Overall Accuracy =  $\text{sum}(\text{diagonal}) / \text{sum}(\text{confusion matrix}) * 100$
- Accuracy for each class =  $(\text{confusion matrix}[i, i] / \text{sum}(\text{row } i)) * 100$
- Average Accuracy = avg (classes accuracies).

### **iii. Predict and Evaluate**

- Given trained model, x test and y test
  - Predict x test using trained model.
  - pass predictions and actual
- to method that calls Build Confusion Matrix and Calculate Accuracy

## 4.8 Results

Adding 2 fully connected layers improved our accuracy.

We then fine-tuned our model by retraining the exit flow of Xception but it did not improve our accuracy by a good margin.

after experimenting/trying different architectures of different model with updating its layer's parameters or add some new layers to fit our classification problem these are the accuracy that we got from each model

Model	Average accuracy	Total Accuracy
VGG16	64.52	67.5
Xception	68.3	71.45
Nasnet	63.91	61.16

Table of result

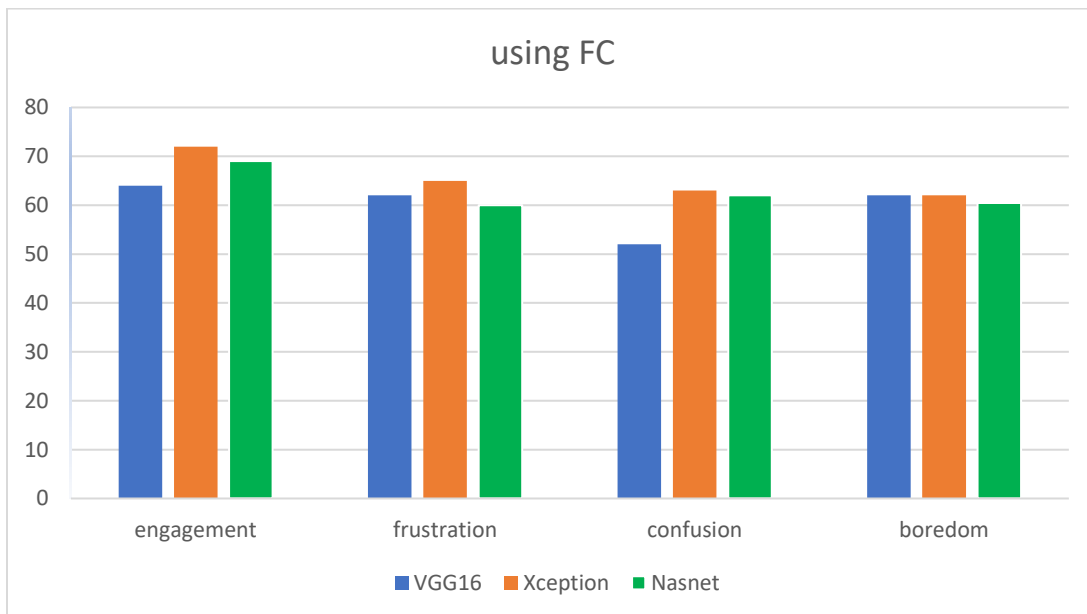


Figure 21 Accuracy

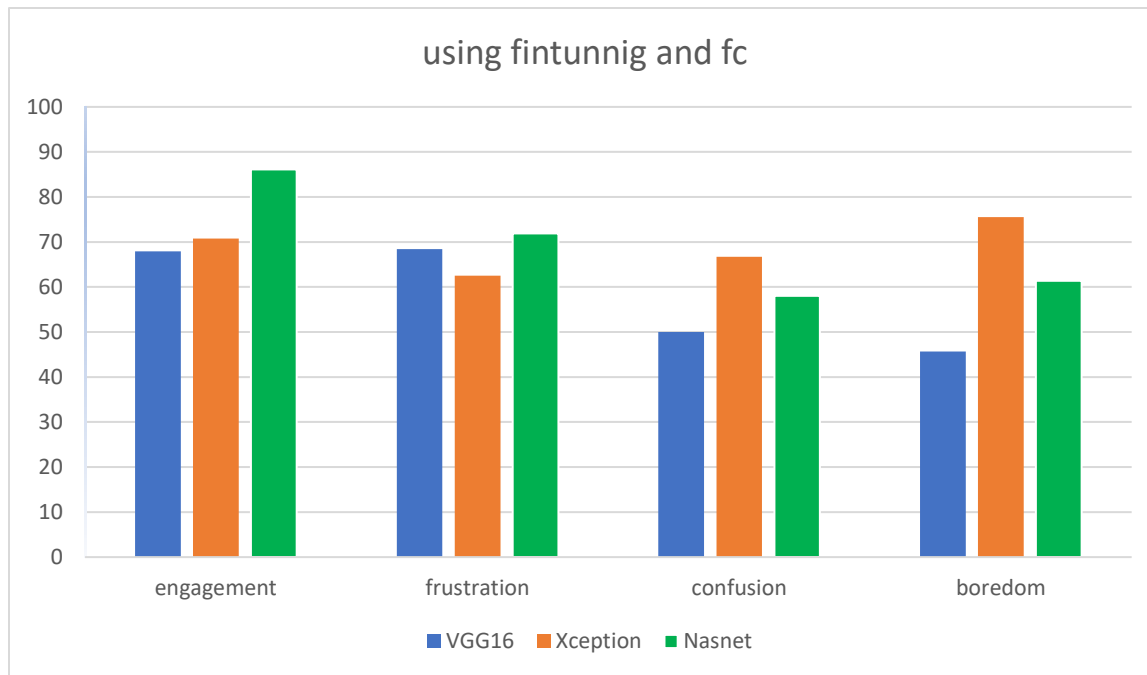


Figure 22 different way accuracy

## 4.9 Best Result

After trying different models and different architectures we have found that Xception is the best model fitting the data.

Xception has 68.3% as average accuracy  
And 71.45 as overall accuracy.

## Chapter 5

### User Manual


#### 5.1 steps for Running project:

##### 1- open mobile application





*Figure 23 splash screen*

## 2- Choose between student and Doctor:



The illustration shows a classroom or library environment. A male teacher in a yellow shirt is pointing at a whiteboard. Two female students are present: one is sitting on a stack of books, and the other is standing next to a stack of books. There are several stacks of books, a potted plant, and a speech bubble coming from the teacher.

 **Continue as student** ☐

 **Continue as Doctor** ☐


**Next** 

Figure 24 student or doctor



### 3- Signups create new account:

8:38 PM | 0.0KB/s

E-learningApp

Here's  
your first

Name

Email

Password

CREATE ACCOUNT!

←

Figure 25 make new account.

## 4- Login

8:36 PM | 44.8KB/s

E-learningApp

**Already have an Account?**

Email  
mohamed

Password  
.....

[Forgot password?](#)

**LOGIN**

[New user? Register Now](#)

+

Figure 26login

## 5- Choose and Upload video to server:

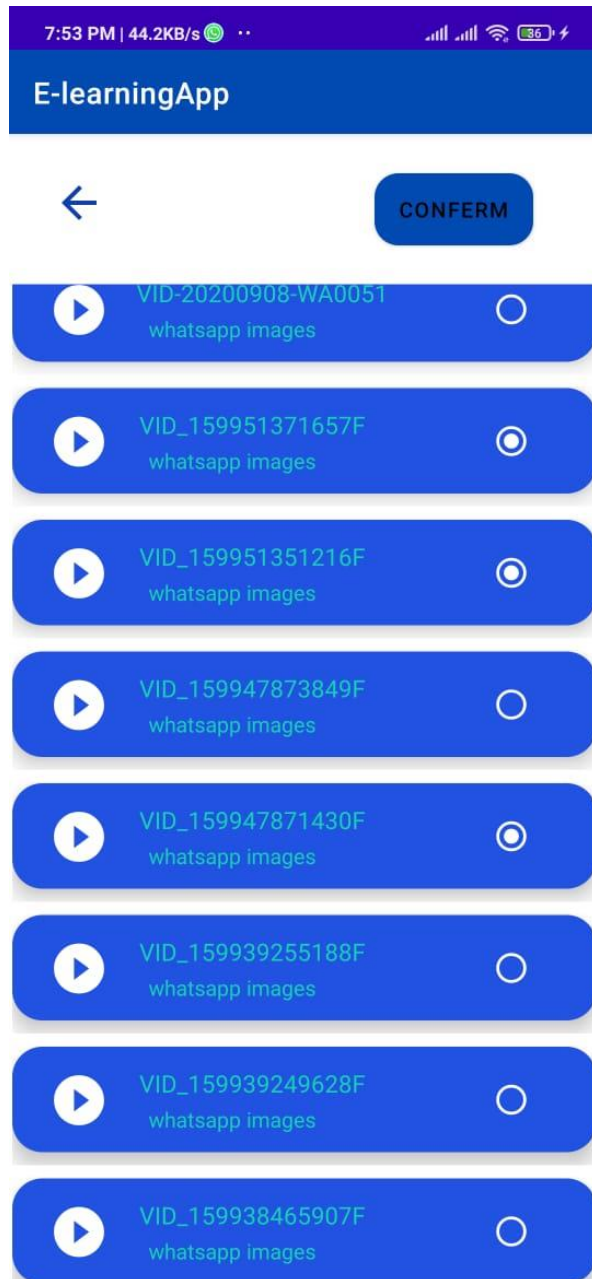




Figure 27 student choose video to upload

## 6- Doctor set interval:

8:36 PM | 1.9KB/s |  42

**E-learningApp**



**SHOW ENTRVALS**

**CONFIRM**

Figure 28set intervals.

## 7- Doctor can see the result:

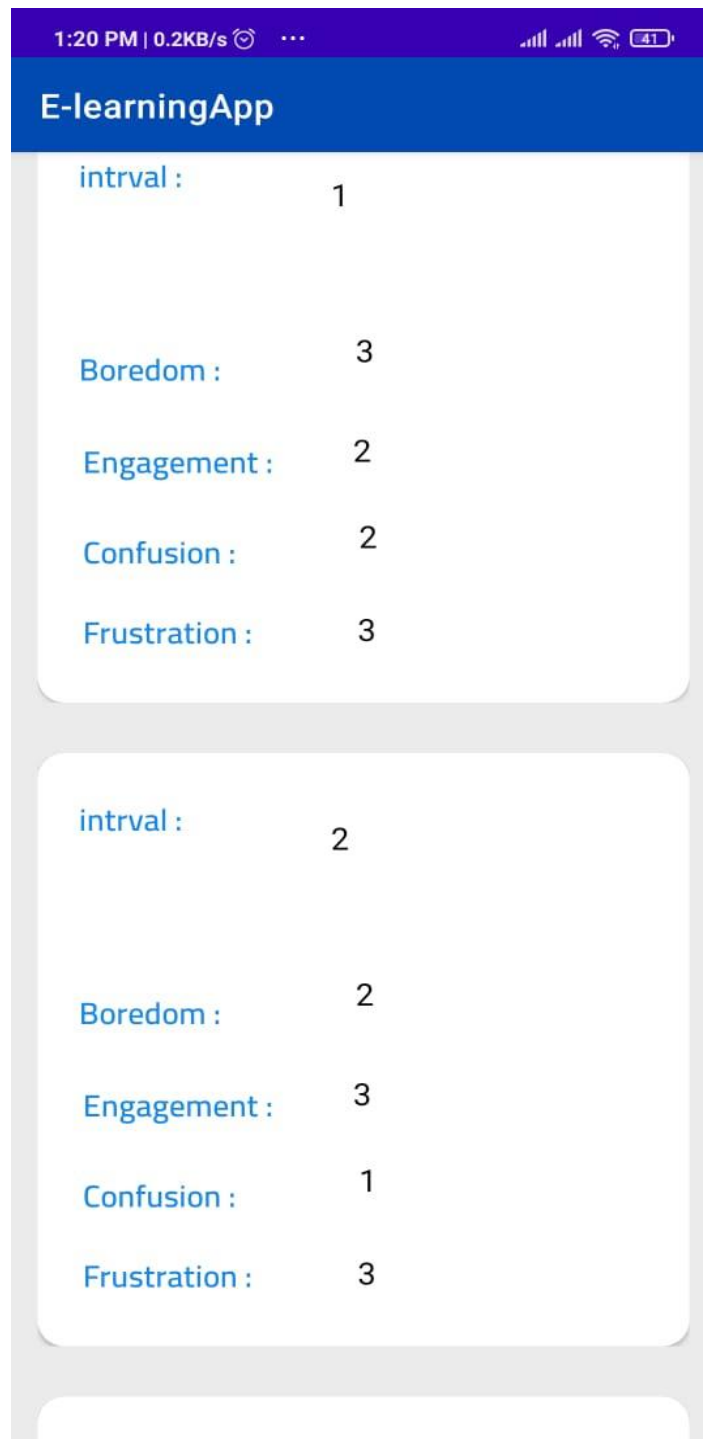


Figure 29 show result

## **Some Notes:**

1-Frame Skip Rate number expresses that 7 frame has been chosen for every 10se.

2-The inputs in form of a Video which has been recorded for one person only.

3-the output then will be a percentage for each expression, the highest percentage will be the dominant class.

## **➤ The Libraries Must be installed:**

Flask Library: Library for supporting API.

Adv:

- build in development server and fast debugger.
- integrated support for unit testing.
- RESTful request dispatching.

## Chapter 6

### Conclusion and Future Work

#### 6.1 Conclusion and Future Work:

In this work, we present DAiSEE, a dataset for user engagement in the wild.

The novelty of DAiSEE comes from the rich information that it has of different affective states such as engagement, boredom, confusion, and frustration.

Each affective state is categorized from very low to very high (without neutral) and is annotated using “wisdom-of the-crowd”.

The dataset captures the nature of real-world e-learning environments in an organic manner, with varying user poses, positions and background noises typically observed in such settings.

It is unique as this is the first publicly available dataset to study these four affective states compared to the seven basic emotions and is the largest available facial emotion/expression dataset for the research community to work on.

We also present benchmarking results for DAiSEE and establish a baseline for the research community to build on.

To help create a more open community for DAiSEE, we present its raw annotation data for conducting research to improve vote aggregation algorithms or for using these annotations in the training process to improve upon the baselines shared in this work.

Going forward, methods that determine geometric features (such as facial fiducials), facial action units, body, and head pose.

gaze and gesture can be used as input to models.

that learn to recognize user engagement.

Also, systems that use such mid-level cues, such as pose and gesture, can often be used to develop cognitive models of the subject which include engagement, attentional focus, and intention.

We hope that DAiSEE assists teachers, content creators and students in the domain of e-learning, advertisement makers, medical professionals, and autonomous vehicle companies in creating better and more responsive systems to help improve human-computer interaction.



## **Tools:**

- Python
- Java
- Google Collab
- Android Studio
- PyCharm
- Python packages
  - NumPy
  - Pandas
  - Sklearn
  - TensorFlow
  - Keras framework
  - ffmpeg
  - SciPy
  - matplotlib

## References

- [1] <https://bdtechtalks.com/2019/08/05/what-is-artificial-neural-network-ann/>, "What are artificial neural networks (ANN)?," p. 1, 3 2 2021.
- [2] KISHAN MALADKAR, "Kind of Ann".<https://analyticsindiamag.com/6-types-of-artificial-neural-networks-currently-being-used-in-todays-technology/>.
- [3] Muneeb ul Hassan, "VGG16 – Convolutional Network for Classification and Detection".<https://neurohive.io/en/popular-networks/vgg16/>.
- [4] D. Nair, "NASNet - A brief overview".
- [5] F. Chollet, "Xception: Deep Learning with Depthwise Separable Convolutions".[https://openaccess.thecvf.com/content\\_cvpr\\_2017/papers/Chollet\\_Xception\\_Deep\\_Learning\\_CVPR\\_2017\\_paper.pdf](https://openaccess.thecvf.com/content_cvpr_2017/papers/Chollet_Xception_Deep_Learning_CVPR_2017_paper.pdf).
- [6] A. M. L. M. A. D. Amanjot kaur.[https://openaccess.thecvf.com/content\\_cvpr\\_2017/papers/Chollet\\_Xception\\_Deep\\_Learning\\_CVPR\\_2017\\_paper.pdf](https://openaccess.thecvf.com/content_cvpr_2017/papers/Chollet_Xception_Deep_Learning_CVPR_2017_paper.pdf).
- [7] J. Brownlee, "CNN," <https://machinelearningmastery.com/how-to-develop-convolutional-neural-network-models-for-time-series-forecasting/>, p. 1.
- [8] o. Hagege.<https://www.dlology.com/blog/how-to-choose-last-layer-activation-and-loss-function/>.
- [9] Google, "firbase".<https://firebase.google.com/>.
- [1 A. Biswal.[https://www.simplilearn.com/tutorials/deep-learning-tutorial/deep-learning-0\] algorithm](https://www.simplilearn.com/tutorials/deep-learning-tutorial/deep-learning-0] algorithm).
- [1 VGG16.<https://github.com/pytorch/vision/blob/master/torchvision/models/vgg.py>.
- 1]
- [1 <https://paperswithcode.com/paper/xception-deep-learning-with-depthwise>.
- 2]
- [1 m. alen, "Nasenet".<https://www.programmersonsought.com/article/62143027595/>.
- 3]