Project Name: ***Oculus Vigilis:Real-Time Attention Model***

Students
Ilke Kas - ixk238

Table of Contents

1. Abstract

In today's world of Internet and increased usage of electronic displays and devices has resulted in increasing the attention related issues like hyperactivity and ADHD, our model addresses this challenge by deploying a real-time attention detection model which utilizes facial recognition technology to monitor and give feedback on various user attention levels. Our project 'Oculus Vigilis" aims at improving engagement of individuals in front of the screen in online educational and professional environments.

We have created our dataset, which comprises video recordings of various individuals engaged in computer interactions, (activity was provided to them), which is annotated with attention scores at five-second intervals. Various features are extracted from these videos such as iris pose, face pose, eye aspect ratio and lip distance by using MediaPipe library. After we labeled these extracted features as one of three attention levels low (1), medium (2) and high (3), we performed data analysis on the dataset we created to see whether it is suitable to train the model. After the data analysis, we trained a Long-Short Term Memory (LSTM) model and achieved 85% accuracy. However, the model could not successfully classify the middle attention score due to imbalance of the middle attention data. We also implemented a User Interface to enable users to interact with our model. In this user interface, users can start, stop and see the video recorded and see the real-time (4-5 second latency) feedback of attention score and program creates basic time vs attention score graphs of the user at the end of the session and provide it as a pdf to the user. This study has many applications from online education to marketing strategies. For further works, the attention score of the multiple individuals can be calculated using this model.

2. Introduction

Developments in technology and digitalization of the world cause people to spend more time with technological devices such as computers, tablets etc. According to the statistics, an American spends 7 hours looking at a screen each day [1]. There are studies indicating that the exposure to the screen is directly associated with self-perceived attention problems such as ADHD and hyperactivity depending on the dose of the exposure [2][3]. This situation creates a loop between the screen time and the attention level of the people. More screen time people are exposed to, the more their attention decreases and when they have to work or study through computers, it is getting harder for them to focus on the work. Most of us experienced this attention deficiency during COVID while using the online meetings and online learning platforms.

In the literature, there are several existing solutions to keep the engagement of the people in front of the computer screen. These solutions can be given like this:

- **Facial Expression Analysis**: Facial expression analysis algorithms can be employed to detect subtle changes in facial expressions that may indicate a participant's level of attention or engagement [4][5][6][7][8].

- **Voice Analysis:** Voice analysis techniques can be used to detect changes in tone, pitch, or volume, which may indicate fluctuations in attention or engagement [9][10].

- **Aspect-Based Sentiment Analysis:** Solutions like Symbl.ai leverage aspect-based sentiment analysis to analyze voice and video conversations, identifying key aspects and sentiments expressed by participants. This can indirectly indicate attention and engagement levels [11][12][13][14].

- **Awareness Tools:** Platforms like Aware use various data points such as keyboard activity, mouse movements, and application focus to infer a participant's attention level during virtual meetings [15][16][17][18][19] .

However, as one can observe, none of the existing solutions did not meet the need for real-time attention feedback from images. So, in this study, we have proposed to implement an advanced version of it which detects the attention level of the individual.

“Oculus Vigilis” has one main objective:

- It aims to help people to overcome their self-perceived attention problems such as ADHD or hyperactivity by giving a real-time feedback of the attention levels of one user using their camera records, especially during online meetings and learning processes.

For the further improvements, we want to give real-time attention level feedback of multiple people for collective events.

As mentioned, “Oculus Vigilis” aims to provide a real-time attention model that detects the attention level of the person in front of the screen. This attention model is trained on the dataset that is created by the team. The dataset is composed of different videos of at least 8 different people recording range from 5 minutes to 2 hours long. The team labeled these dataset for every 5 seconds as one of these: High, Medium. Low attention levels.

While training this model, the various extracted features in the dataset preparation part are used. Eye aspect ratio (EAR), lip distance, face pose, and iris pose are computed for each frame by using the facial landmarks and saved as features in the dataset. These data are given as an input to our Long-Short Term Memory attention model.

This project can be used in many settings such as online meetings, online learning platforms and online advertising optimizations. In online meetings, it can be hard to focus on the presentations or lectures while people are already in their comfort zones. This kind of attention model helps the presenters in the meeting to keep the attention levels high. For example, this attention model can be integrated with online meeting platforms such as zoom. When the overall attention of the class decreased, the meeting platform may create a feature for the presenter to alert people.

Similarly, in online learning platforms, when the attention level of the learner decreases, the platform may send an alert to the user itself or pause the video/learning material so the user will not miss any kind of information important for the learning process. Finally, this model can be used in online advertisement optimizations to maximize the engagement and conversion rates by identifying the patterns of the user's attention.

This project is expected to enhance the learning and productivity through technological devices and online applications, improves the user experience of the online meetings and helps people to address their attention related issues such as ADHD and hyperactivity.

We will mention the methodology used to develop “Oculus Vigilis” in **Section 3.** After that, in **Section 4,** we will share the results and discuss the results. Finally, In **Section 5,** we will conclude the project. In **section 6,** the work division between the team members will be mentioned.

3. Methods

In order to accomplish the project objectives, we used the “Software Development Cycle (SDLC)” methodology. This methodology enables the project to go through several stages as the team adds new features and fixes the bugs [20]. As a high-level approach to this project we can mention the requirement analysis, design phase, development, testing.

3.1. Requirement Analysis

3.1.1 Functional Requirements

- The user is able to start a video camera and see herself/himself
- The user is able to see the real-time attention feedback score of her/him
- At the end of the session, user is able to see the overall attention score in a plot time versus attention score

Functionality of the application can also be seen in **Section 3.2.2 Use Case Diagram** in **Figure 2.**

3.1.2 Nonfunctional Requirements

- **Usability-** The user interface is simple and provides quick access to essential features of the application.
- **Privacy-** The collected personal data from the users such as their camera records only used for attention tracking. These personal data were not processed for any other purposes which are incompatible with the main purposes of the project. The user’s face was not kept in the database/project.
- **Performance-** The project is a real-time system which processes the faces of the user and presents the current attention level. Besides that, the attention level of the presentation should have a higher accuracy.
- **Extensibility-** The design and implementation of the system did not hinder future needs and updates of the system such as real-time attention analysis for multiple users. The functionality of the project can be expanded in the future.

3.2. Design, Implementation and Test Phases

This project needed a detailed design in order to make implementation possible. Therefore, in this section, the proposed software architecture is explained in a detailed way. Software architecture is a fundamental structure that plays an integral role in understanding software systems and providing a road map for their development. This section presents a brief overview of the software architecture’s key components including subsystem decomposition.

3.2.1 Subsystem Decomposition

Subsystem decomposition is a significant aspect of software architecture that divides systems into smaller subsystems to create more manageable components. By dividing the system into smaller subsystems, the process of development becomes more manageable and testable. Then, subsystems are integrated to form the complete system.

The purpose of the view is to demonstrate the main functionalities of the application. Hence, the system is divided into three subsystems and each of them has different responsibilities. In the diagram, the main subsystems and functionalities can be understood.

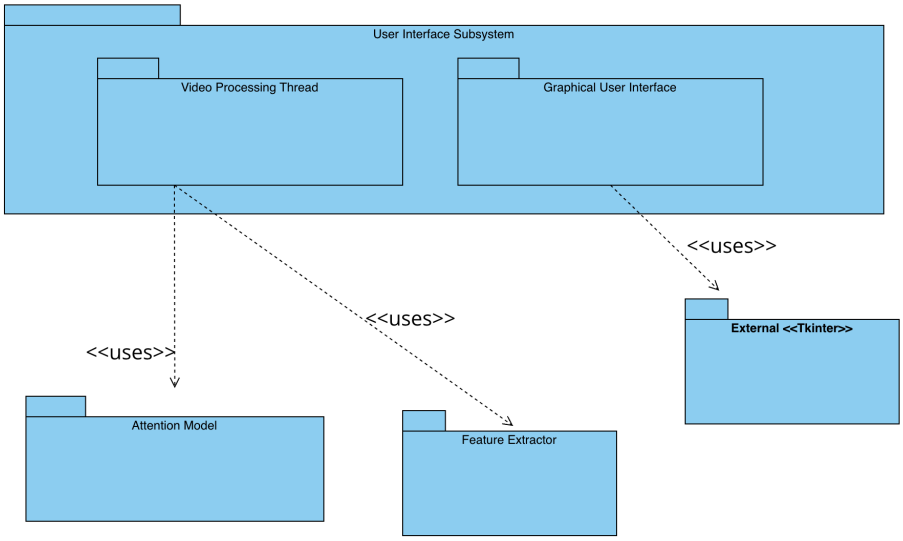


Figure 1- Subsystem Decomposition of Program

As one can see from the subsystem decomposition of the project in **Figure 1,** User Interface Subsystem is responsible for various functions such as starting/stopping the application, visible real-time attention score feedback and camera stream. This subsystem is using two threads, video processing thread and the graphical user interface thread (main thread). Video Processing Thread is taking the video of the user and by using Feature Extractor subsystem, extract the necessary features to assess the attention of the user. It also uses Attention Model to give real-time feedback to the user. Graphical User Interface uses an external Tkinter module in order to supply basic input/output interface devices such as buttons, labels, windows etc. Let's go over subsystem based methods, implementation and design.

3.2.1.1 Attention Model

3.2.1.1.1 Dataset Preparation

In order to train a neural network for attention classification, an initial requirement was a labeled dataset. In order to find the dataset we searched the available datasets, but we could not find one that can help us. Therefore, we decided to create our own dataset. We asked our friends to save around 2 hours of video of themselves by following the directions we provided by "video_directives" we created for the subjects [21]. The subjects are wanted to sign the informed consent and media release forms before recording their videos. One can find the original copy of these forms in **Appendix A** and **Appendix B.**

Initially, we decided to make predictions for video chunks that were 5 seconds in duration. Consequently, we implemented two Python scripts. In the first script, extractor.py, we are extracting various facial features from a sequence of frames, such as Eye Aspect Ratio (EAR), Lip Distance, Face Pose, and Iris Direction. We do this by finding the facial landmark coordinates by using the Mediapipe FaceMesh model [22].

We calculated the eye aspect ratio from the eye landmarks and used it directly for predictions. The lip distance is calculated using landmarks on the upper and lower lips. A threshold is set, and if the lip

distance is greater than the threshold, the result is saved as 1 (open); otherwise, it is saved as 0 (closed). OpenCV is used to solve the perspective-n-point (PnP) problem and estimate the face pose. Once again, a threshold is set to determine if the user's face is facing the screen or not. Based on this threshold, the result is saved as 1 (facing the screen) or 0 (not facing the screen). For the iris pose, we are using iris landmarks of FaceMesh and again there is a threshold. If irises are looking on the screen the result is saved as 1 (on screen), else 0 (out screen).

Second script we used to prepare the dataset is labeling.py. Labeling.py is using extractor.py in order to extract the features and label data. In labeling.py, the selected video was divided into 5-second-long chunks, and each chunk was displayed on the screen individually. Group members used the keyboard to assign labels to these video chunks, categorizing them into three classes: Low, Middle, and High. The assigned labels were then stored in a CSV file together with the extracted features. Each group member was responsible for labeling a video duration of nearly 4 hours. Ultimately, we obtained a labeled video dataset that is 29.390 seconds (8.2 hours) long. The head of labels.csv can be seen in **Figure 2**. Chunk index indicates the order of the video chunk in recorded video, label indicates the attention: 1 for low, 2 for middle, 3 for high attention. One can find the dataset in the submitted “labels.csv” file.

	video_name	chunk_index	frame	ear	lip_distance	face_pose	iris_pose	attention_score
0	1_1	0	0	0.416018	0	1	1	3
1	1_1	0	1	0.412338	0	1	1	3
2	1_1	0	2	0.426329	0	1	1	3
3	1_1	0	3	0.425512	0	1	1	3
4	1_1	0	4	0.416889	0	1	1	3
...	...	...	...	...	...	...	...	...

Figure 2- Dataset Features and Some Rows

After receiving the videos from the subjects, we divided these long videos into 5 minute long videos. In this way, our program did not raise an out of memory exception while extracting features. “Video_name” refers to the video that extracted the features. For example, the video name “1_1” refers to the first person’s first 5 min video part, “4_2” refers to the fourth person’s second 5 minute video part, etc. As you can see, there are 8 different faces used in this dataset. “Chunk_index” refers to the 5 second parts of each 5 minute video. Each chunk_index has 100 frames saved to the dataset. The features are extracted from these frames by using extractor.py. All 100 frames belonging to the same chunk index have the same label since we label each chunk, not frames.

3.2.1.1.2 Dataset Analysis

After preparing the dataset, we analyzed the dataset by using basic analyzing methods such as covariances, correlations, histograms, box plots and scatter plots. This part is important since we created the dataset, we need to check whether the data is suitable to classify for the attention score and understand the relation between the features.In order to understand the relationship between features, we calculated the covariance and correlations of the features by using cov() and corr() functions of the pandas library. Both of these measures show the linear relationship between the features. Interested readers can see the covariance and correlation definitions in **Appendix C.** In addition to these, histograms, box plots and scatter plots are used to analyze the data. It is important for the dataset to be balanced in order to classify any data with high accuracy. Otherwise, the statistical or the machine learning models have a tendency to classify the test data as the most occurring class. Histograms can help us to decide whether this dataset is balanced or not in terms of the labels.

Box plots are used to compare the distribution of the different data groups in general. In this study, box plots are used to compare the distributions of features in three different attention score classes “1”, “2” and “3”. In this way, we can analyze whether the value of any feature can affect attention score class significantly by looking at the box plots. We can interpret some feature importance results by looking at these plots. In order to draw the boxplots, I used the boxplot function of the pandas library.

Scatter plots are used to observe the relationship between two numeric variables. In this study, scatter plots are used to visualize the relationship between variables who have strong correlation values. In order to draw the scatter plots, I used the plot.scatter function of the pandas library.

3.2.1.1.3 Model Training

After preparing the dataset and analyzing it, data is preprocessed before model training. Since we are going to train an LSTM (Long-Term Short Memory) which is an RNN (Recurrent Neural Network) model, we need to create the sequential data in the proper way.

The reason RNN is used is that video data is considered as the sequence data and normal neural networks cannot handle this kind of data. RNN defines recurrence relation between time sequences. In this way, the data from previous time steps can influence the prediction of after steps. However, during the back propagation of the basic RNN model, issues like exploding or vanishing gradient problems can occur depending on the values in weight matrices. In order to avoid this, LSTM (Long-Short Term Memory) models are developed. LSTMs basically consist of gated cells to track information throughout many time steps. These gates can forget, store, update and output the information. The overall structure of LSTM cells can be seen in **Figure 3:**

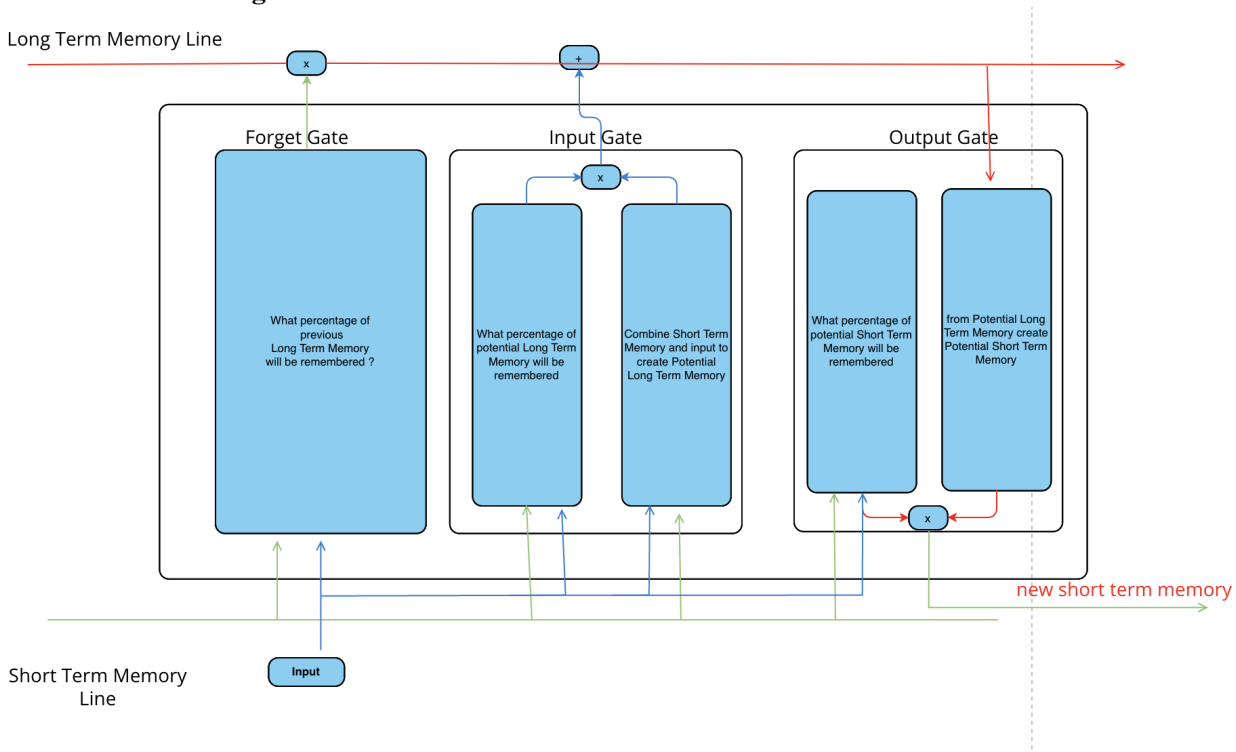


Figure 3 - The Architecture of an LSTM Module

As seen from **Figure 3,** the LSTM cell consists of 3 gates, Forget, Input and Output [23]. Forget cells taking input and short term memory from previous cells and multiply them with corresponding weights and add biases before sending them to the Sigmoid function [23]. Sigmoid function returns a number between 0 and 1 and multiplies it with the previous Long Term Memory [23]. This deals with “What

percentage of previous Long Term Memory will be remembered?[23]". Input gate has two main parts: One part is creating a new long term memory by combining the past short term memory and input [23]. It combines them by multiplying the corresponding weights and adding bias before taking the tanh of it. The second part calculates again "What percentage of potential Long Term Memory will be remembered?"[23]. Its structure is the same with the forget part except this time output of it will be multiplied with the newly created long term memory and added to the long term memory line [23]. Finally, output gate calculates the potential short term memory and the percentage to be used of newly created short term memory [23]. To calculate the percentages, LSTM uses sigmoid activation function while to calculate potential memory, it uses tanh activation function.

We created sequences which are the grouped version of the dataframe in **Figure 2** in terms of the same video_name and chunk_index features. In other words, we stack together 100 frames that belong to the same chunk (5 seconds long video parts).

```
(            ear  lip_distance  face_pose  iris_pose
457500  0.060190             0          1          1
457501  0.060162             0          1          1
457502  0.035067             0          1          0
457503  0.035067             0          1          1
457504  0.056887             0          1          1
...          ...           ...        ...        ...
457595  0.096494             0          1          1
457596  0.094293             0          1          1
457597  0.099405             0          1          1
457598  0.096849             0          1          1
457599  0.094581             0          1          0

[100 rows x 4 columns], 0.0)
```

Figure 4- Created Sequence from Dataset

Figure 4 shows an example of one sequence (5 seconds video frames) with label 0. Since we are going to make classification, when labeling we label attention score 1 as 0, 2 as 1 and 3 as, during training. It can be thought of as encoding of the attention scores.

After data is created with these sequences and splitted into test and train sets by 0.20 proportion, the attention model is created. Prediction model is built with three LSTM layers with 256 units. Dropout layers with a rate of 0.75 are inserted after each LSTM layer to decrease overfitting. The final layer is a dense linear layer with 3 units representing the three classes: Low, Middle, and High. **Figure 5** shows the mentioned architecture of the attention model. After the model is created, the Attention Predictor takes the maximum argument between these three classes and predicts that value as the attention score. The model is trained on the A100 GPU that Google Colab provided.

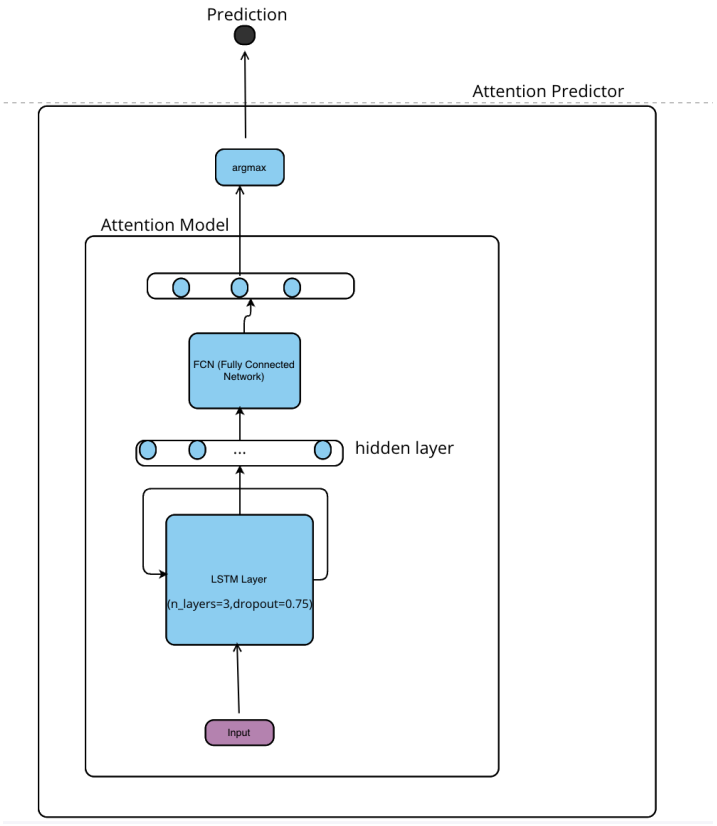


Figure 5- Attention Model Architecture

The model is compiled with the AdamW optimizer, using a learning rate of 0.0001. The loss function is specified as cross-entropy which is the most suitable loss function for multi-class classification. The model is trained with the training data, validation split of 20%. Number of epochs is set to 250 and batch size is set to 64. After training, the model is evaluated using the test data.

3.2.1.2 User Interface Subsystem

User Interface utilizes several libraries in order to interact with the user through buttons, images etc. “Tkinter” is the main library used for basic input output items such as buttons and labels in UI. This part uses a UI.py script that has two threads. Main threads, keeps the GUI updated such as the functions of buttons, visibility of video recording, average score label etc. On the other hand, video threads capture the video of the user, extract the features from the video frames and use an attention model to predict the attention score of the user.

3.2.2 Sequential Diagram

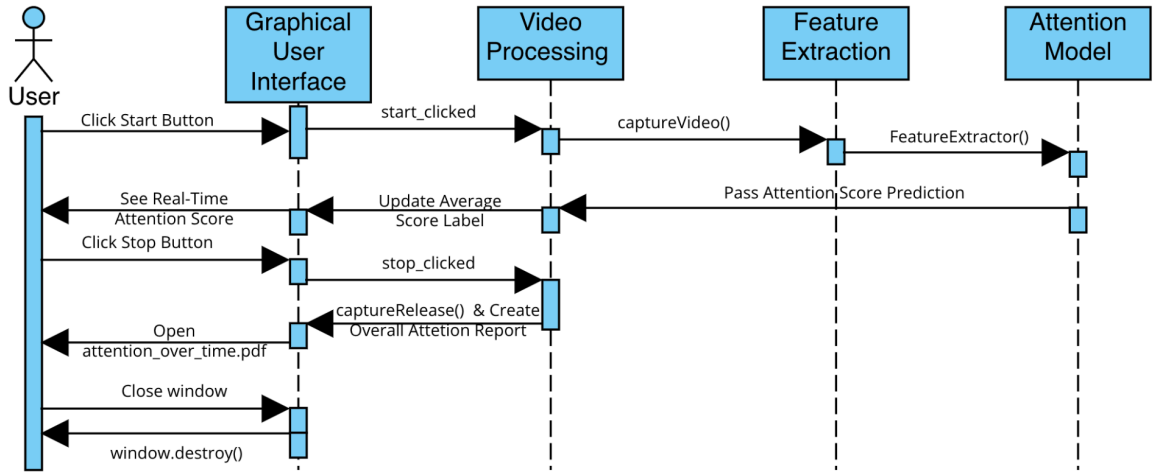


Figure 7- Sequence Model for User

Figure 7 shows the sequential diagram of the project. The user starts by clicking the "Start Button" in the graphical user interface, which initiates the video processing. After video is captured, features are extracted and sent to the attention model to predict the attention score. The system then displays both the video of the user and the attention score of the user. The user can click the "Stop Button" to end the process and see attention_over_time.pdf which shows the plot of time vs attention score of the user during the session. Finally, the window is destroyed and the session completed when the user clicks the "x" on the right top of the window.

4. Results and Discussion

4.1. Results and Discussion of Dataset Analysis

4.1.1. Covariances and Correlations

	ear	lip_distance	face_pose	iris_pose	attention_score
ear	0.106273	0.065055	0.113666	0.065948	0.068942
lip_distance	0.065055	0.075681	0.089596	0.039329	0.022941
face_pose	0.113666	0.089596	0.310015	0.142438	0.195395
iris_pose	0.065948	0.039329	0.142438	0.203381	0.230746
attention_score	0.068942	0.022941	0.195395	0.230746	0.932120

Figure 8 - Covariances

	ear	lip_distance	face_pose	iris_pose	attention_score
ear	1.000000	0.725391	0.626222	0.448574	0.219046
lip_distance	0.725391	1.000000	0.584929	0.317003	0.086372
face_pose	0.626222	0.584929	1.000000	0.567254	0.363485
iris_pose	0.448574	0.317003	0.567254	1.000000	0.529959
attention_score	0.219046	0.086372	0.363485	0.529959	1.000000

Figure 9 - Correlations

Figure 8 shows the covariances between the features while **Figure 9** shows the correlations between the features. Ear feature means “Eye-Aspect-Ratio” as mentioned.

Figure 8 Discussion: Since covariances are not standardized they are scale dependent. Therefore, we cannot assess the direct relationship between two variables by looking at covariance values only. We can assess the direction of their relationship though. The reason behind this is the fact that these features can take larger values compared to other ones and covariance does not scale it. Besides that, it is not easy to analyze the relationship between independent and dependent variables by looking at the large tables above. Negative covariance values implies that the higher values in one variable tend to correspond to lower values in another variable while positive covariance values implies higher covariance value in one variable tends to correspond to higher values. Let’s analyze the covariance values between the features under the lights of this information.

All of the features have positive covariance values that means the direction of their relationship is the same. The red values show minimum and maximum covariance values in each row.

Figure 9 Discussion: Correlation gives more information to us between the relationship of the variables since it does not depend on the scale of the features. In other words, it uses standardization. According to this figure, attention score depends on the iris pose most compared to the other features since the correlation value is higher than the others 0.53. After iris pose, the face pose has a stronger relationship with attention score compared to the ear and lip distance features. Finally, the feature that has a weaker relationship between attention score is considered as lip distance. This analysis can give us insights about how features may affect attention score.

4.1.2. Visualizations

4.1.2.1. Histograms

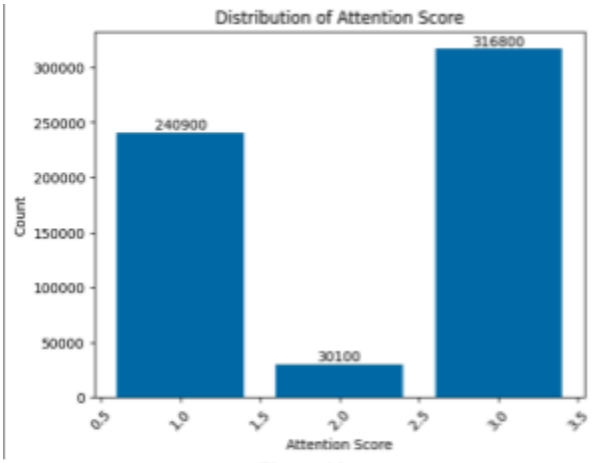


Figure 10

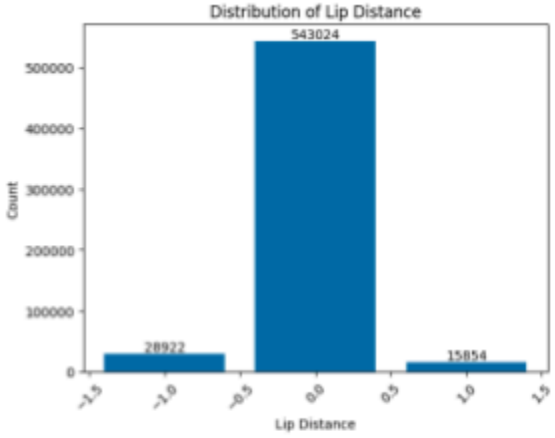


Figure 11

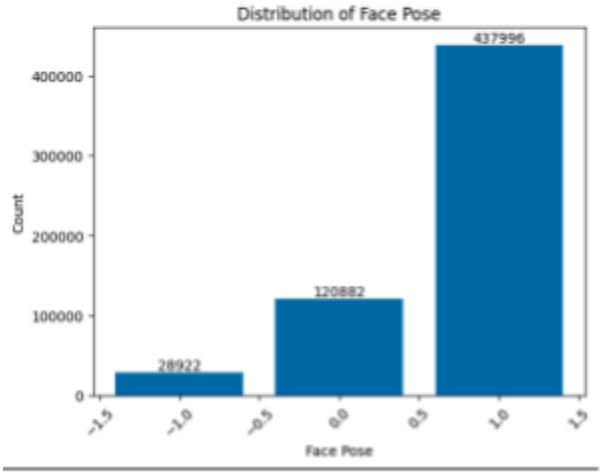


Figure 12

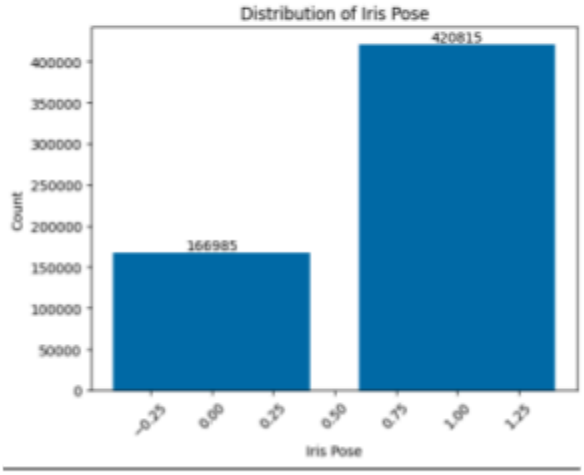


Figure 13

Figure 10-13 - Histogram of Categorical Features in Dataset

Figure 10-13 shows histogram of the features vs the number of frame counts. Attention score can take values 1,2 and 3 as seen from **Figure 10.** By looking at this figure, we can easily see that the dataset is imbalanced in terms of attention score 2 value. This may cause the model to not to classify class 2 correctly. However, since it is an intermediate value, we can ignore that problem. Most of the frames have lip distance 0 according to **Figure 11.** This means that in most of the frames, the subject's mouths are closed. Lip distance -1 means that the facial landmarks could not find the lip landmarks or mouth is covered in some way. **Figure 12** shows that most of the subject's head pose towards the screen (label 1) while some of them are out of the screen (label 0). -1 label in face pose also means that the face landmarks could not be found due to several reasons. Finally, the distribution of iris pose in **Figure 13** shows us that in most of the frames, subjects are looking at the screen.

4.1.2.2. Boxplots

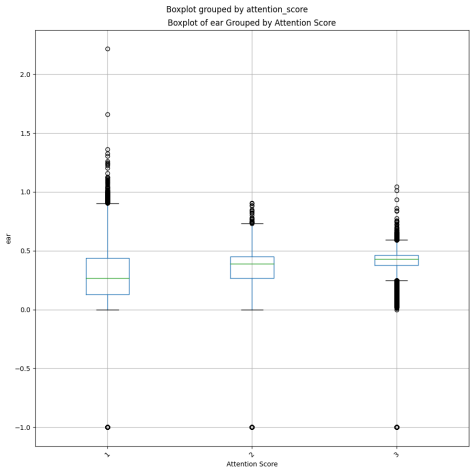


Figure 14- Boxplot of Eye Aspect Ratio Feature Grouped by Attention Score

Since Eye-Aspect-Ratio is the only continuous value in our features, looking at its boxplot rather than histogram makes more sense. **Figure 14** shows the distribution of EAR values in terms of attention scores 1,2 and 3. Medians for each group seem so close to each other for this plot. The interquartile range for each attention score group seems similar and short. However for group 3, it is shorter than the other. This means the data is less dispersed for this feature. The median seems closer to the top of the box, the distribution seems negatively skewed for attention score 2 and 3. That means the median is bigger than the mean. According to this box plot, there is an obvious difference between the different EAR values for attention scores. For attention scores to become high, EAR values are generally higher and not in a variety range. Lower attention scores have lower EAR value compared to high attention scores and have more variety. There are also many outliers that show there are many data which are more extreme than the expected variation. The other boxplots of the features can be seen in **Appendix D.**

4.1.2.3. Scatter Plots

Scatter diagrams are useful to visually evaluate the relationship between pairs of variables.In these plots, strong linear correlation between the features could not be found and since most of the data are categorical, these plots did not reveal any additional information in addition to correlation values. One can find all of the scatter plots in the file “scatterplots.zip” file that I submitted or in **Appendix E**.

4.1.2 Results and Discussion of Attention Model

After training of the data, the best model is saved to the checkpoint logs. The best model has accuracy 0.85 and loss 0.42 as seen in **Figure 15.**

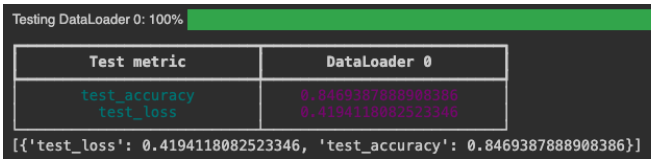


Figure 15- Test Accuracy and Loss of Attention Model

The classification report of the evaluation of the model can be seen in **Figure 16.**

	precision	recall	f1-score	support
0	0.90	0.82	0.86	479
1	0.00	0.00	0.00	61
2	0.82	0.95	0.88	636
accuracy			0.85	1176
macro avg	0.57	0.59	0.58	1176
weighted avg	0.81	0.85	0.83	1176

Figure 16- Classification Report of Attention Model

The meaning of these metrics are explained in **Appendix F,** for the interested reader. As seen from the confusion metrics, The model has a good accuracy around 0.85. When we look at the class 0 (attention_score 1 class), we can see that it has pretty high precision, recall and f1-scores. When we look at the class 2 (attention_score 3 class), it also has a good precision, recall and f1-score. However, when we look at the class 1 (attention_score 2 class), we see that it has 0 precision,recall and f1-score. That means that this model could not classify middle attention. It only can classify the low and high attention scores. Therefore, macro average is quite low compared to other scores. The reason for this is the imbalance of the dataset in terms of the middle attention. We did the labeling in this way: If the subject did not look at the camera, yawn or sleepy during the whole 5 seconds, we labeled it as 1. If the subject looks at the screen most of the time and has attention, we labeled it as 3. Otherwise, if some parts of the 5 second have attention and some parts of the 5 second do not have attention, we labeled it as 5. Because of that, as expected the number of middle attention labeled data is pretty low compared to the low and high attention labeled data. We can also see that in histograms and support value in **Figure 16.** In order to avoid this for the feature work, one can use methods like undersampling or oversampling methods. However, in this study, we did not use such methods since it still reflects the attention score truly.

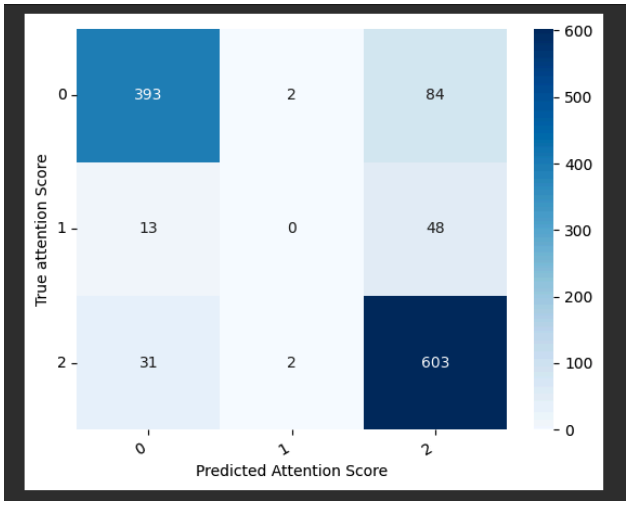


Figure 17- Confusion Matrix of Attention Model

One can also see the confusion matrix of the model in **Figure 17.** As we can see, the model is successful when predicting the low score and high score attention scores. However, it did not classify any of the 61 data as middle attention score. Even though this is not wanted, for the sake of the application it can be ignored. However, in order to have a better model, more balanced data may be needed with more frames with middle attention values.

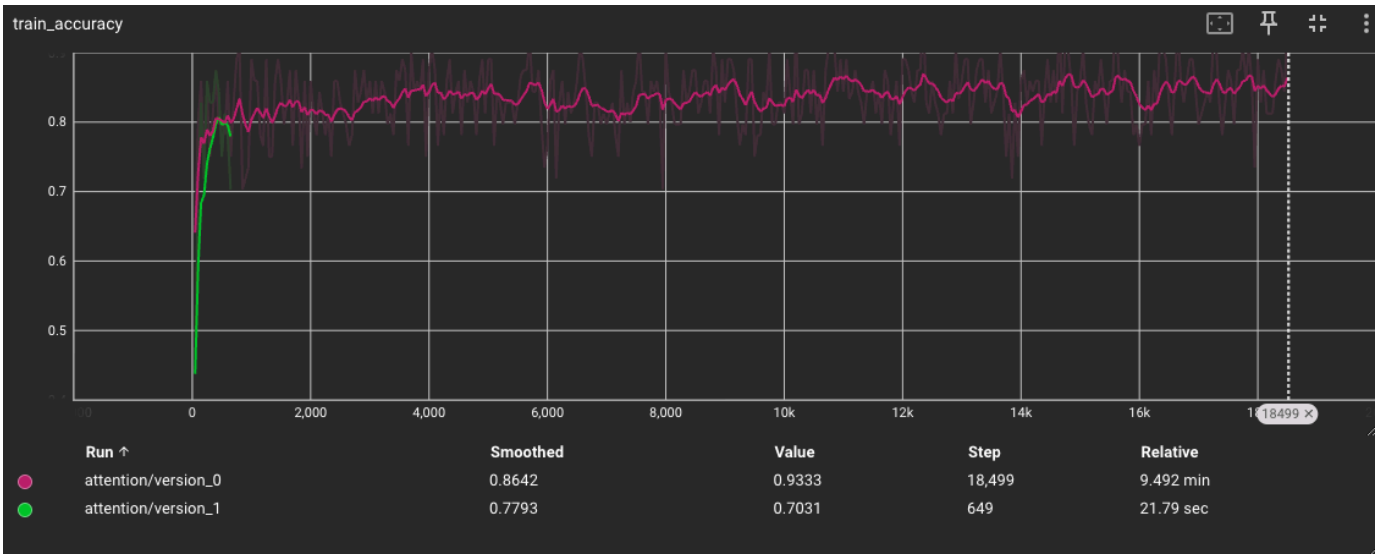


Figure 18- Train Accuracy Graph of Model over Epochs

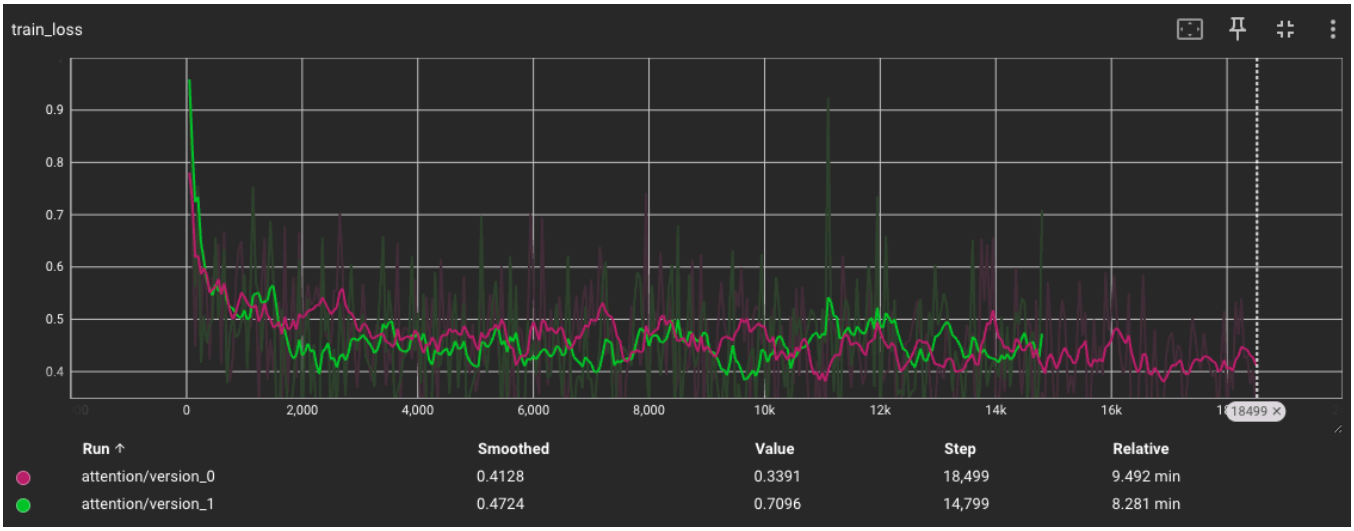


Figure 19- Train Loss Graph of Model over Epochs

Figure 18 and **Figure 19** shows the train accuracy and train loss graphs respectively versus the number of epochs. We can easily see that train accuracy increases and train loss decreases as expected from the model. However, in order to understand whether the model overfits or not, let's look at the validation accuracy and loss in **Figure 20** and **Figure 21.**

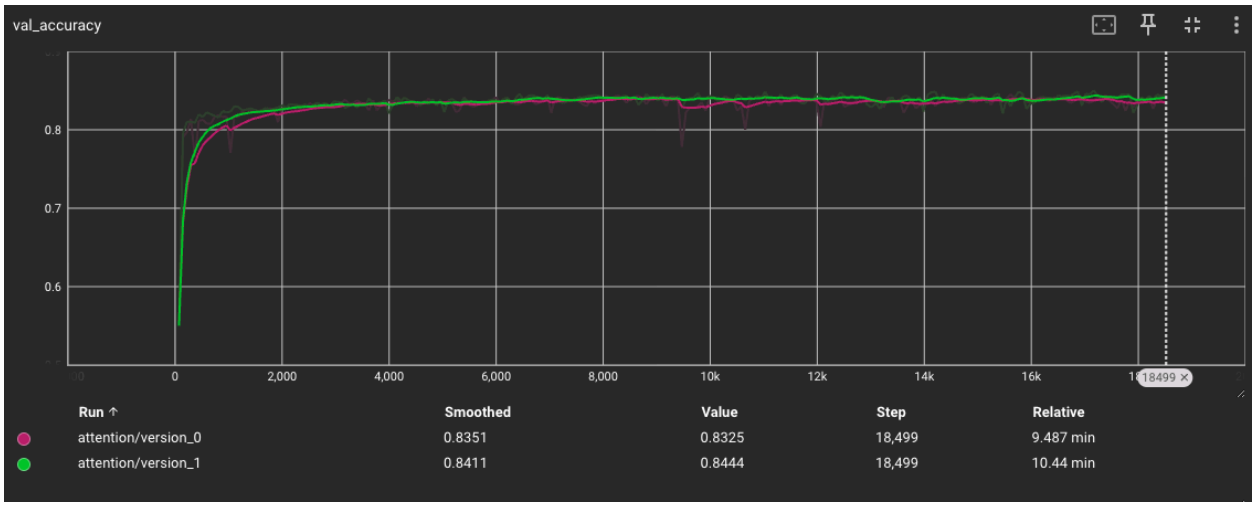


Figure 20- Validation (Test) Accuracy Graph of Model over Epochs

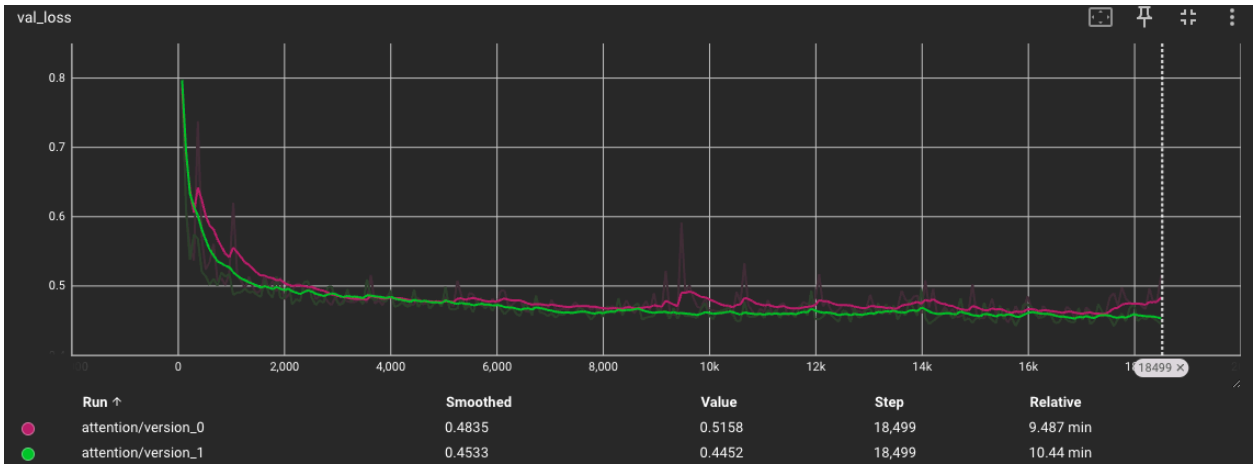


Figure 21- Validation (Test) Loss Graph of Model over Epochs

As we can see from **Figure 20** and **Figure 21,** the model did not overfit since it performs nearly the same pattern on validation data.

Screenshots From the User Interface

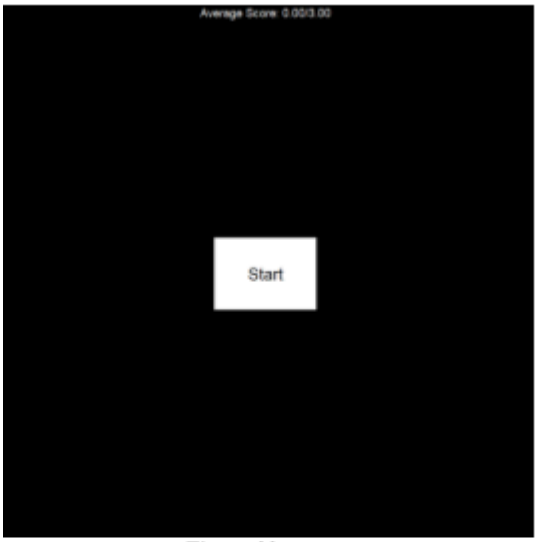


Figure 22

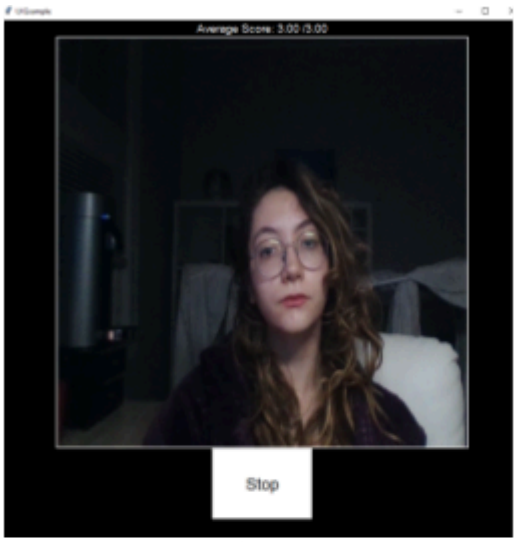


Figure 23

Figure 24

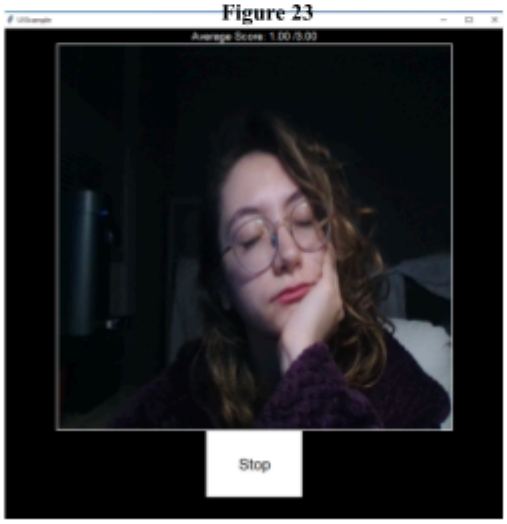


Figure 25

Figure 22-25- Screenshot of User Interface

Figure 22-25 shows some screenshots from the application. As one can see, when the user focuses on the screen, she gets a high attention score of 3.00/3.00. When the user looks away as in **Figure 24,** the attention score decreases. When the user starts to sleep as in **Figure 25,** attention score decreases more.

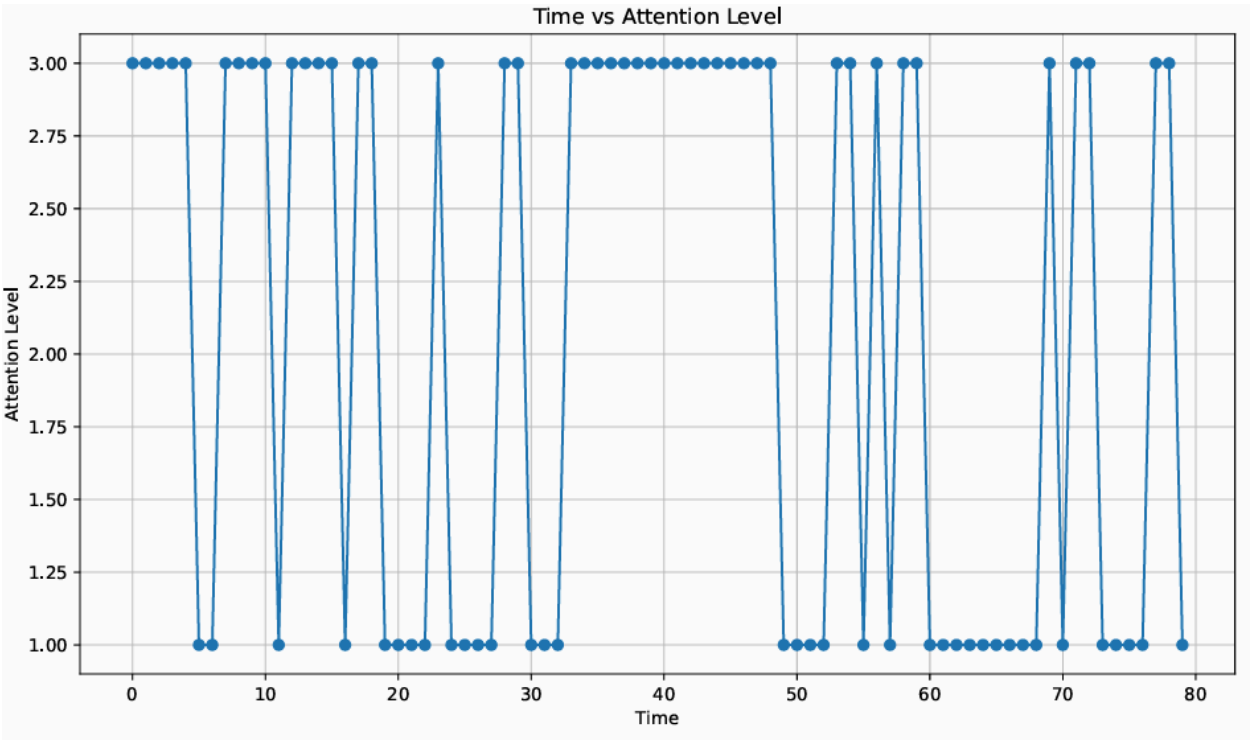


Figure 26- Time vs Accuracy Score Plot After Stopping UI

After stopping the application, the user can see time vs attention score plotted as in **Figure 26** and saved into the same directory of the code as a pdf**.**

5. Conclusion

In "Oculus Vigilis", we aimed to create an attention-detecting model that can calculate the level of attention of students/individuals in an online meeting like zoom/Google Meet/teams etc.

The model's efficiency in online environments especially on educational and various online meeting platforms shows its potential to substantially improve the online interactions and attention of the users in the meeting.

This model is a successful development of a real-time attention model that uses facial recognition technologies and various machine learning technologies to dynamically assess and enhance user engagement.

If we are getting a 3.0/3.0 score in the model it indicates that the individual is fully attentive and if the score is below 2.0/3.0 then it indicates low attention of the user.

If an individual is sleeping then the score is 1.0/3.0 as shown in the screenshots attached above in the report.

Future Developments/Possibilities: Future developments and various modifications can significantly improve the model's overall accuracy and efficiency.

It could include exploration of various additional biometric indicators like pupil dilation, and heart rate to enrich the model's sensitivity to subtle attention shifts.

Model Applications:

- Vehicular Safety systems to monitor driver alertness.
- Medical Settings to assess patient engagement during telehealth sessions.

In conclusion, our project "Oculus Vigilis" not only represents a significant step forward in real-time attention systems but it also serves as a foundational project that sets the stage for the next generation of user interface technologies. By continuing to build on these results, future innovations can deliver more intuitive and responsive digital experiences that are finely attuned to the user's cognitive state.

6. Roles

Team Member	Role in the Group
Ilke Kas	• Data Labeling • Data Analysis • Model Training/Evaluation • UI implementation • Final Report Parts: Methods, Results and Discussion, References, Appendix

7. Notes

- **Code** can be found in the provided link in reference [24].
- **Code for Attention Model** can be found in the link in reference [27].
- **Dataset** can be found in the provided link in reference [25].
- **Videos** used to extract the dataset can be found in reference together with video directives provided for subjects [26].

8. References

[1] R. Moody, “Screen Time Statistics: Average in the US vs. rest of the world,” Comparitech, https://www.comparitech.com/tv-streaming/screen-time-statistics/ (accessed Mar. 6, 2024).

[2] I. Montagni, E. Guichard, and T. Kurth, “Association of Screen Time with self-perceived attention problems and hyperactivity levels in French students: A cross-sectional study,” BMJ Open, vol. 6, no. 2, Feb. 2016. doi:10.1136/bmjopen-2015-009089
[3] J. Wallace, E. Boers, J. Ouellet, M. H. Afzali, and P. Conrod, “Screen time, impulsivity, neuropsychological functions and their relationship to growth in adolescent attention-deficit/hyperactivity disorder symptoms,” Scientific Reports, vol. 13, no. 1, Oct. 2023. doi:10.1038/s41598-023-44105-7
[4] Zhihui Zhang,,Facial expression recognition in virtual reality environments: challenges and opportunities“https://www.frontiersin.org/journals/psychology/articles/10.3389/fpsyg.2023.1280136/full
[5] Bärbel Bissinger,Emotion Recognition via Facial Expressions to Improve Virtual Communication in Videoconferences,”https://link.springer.com/chapter/10.1007/978-3-031-35599-8_10
[6] Peter Eisert and Bernd Girod,Analyzing Facial Expressions for Virtual Conferencing,” https://web.stanford.edu/~bgirod/pdfs/EisertCGA98.pdf ”
[7] Bärbel Bissinger,Emotion Recognition via Facial Expressions to Improve Virtual Communication in Videoconferences
“https://www.researchgate.net/publication/372225905_Emotion_Recognition_via_Facial_Expressions_to_Improve_Virtual_Communication_in_Videoconferences ”
[8] Asanka D. Dharmawansa, Develop a Monitoring Tool and Extract Facial Expression towards the Analyzing Student Behavior in Three Dimensional Virtual Environment,
“https://ieeexplore.ieee.org/document/6031265 ”
[9] “Five9 named a leader in conversational AI by Aragon Research,” Five9, https://www.five9.com/ (accessed Mar. 8, 2024).
[10] Ameyaa Bagwe, DIGITAL CLINICAL VOICE ANALYSIS,”https://www.linkedin.com/pulse/digital-clinical-voice-analysis-ameyaa-bagwe/ “
[11] Amelia.ai, https://amelia.ai/ (accessed Mar. 9, 2024).
[12] Eric Giannini,Using aspect-based sentiment analysis for voice and video conversation with Symbl.ai,”https://symbl.ai/developers/blog/using-aspect-based-sentiment-analysis-for-voice-and-video-conversation-with-symbl-ai/ “
[13] Diaasq : A benchmark of conversational aspect-based ..., https://aclanthology.org/2023.findings-acl.849.pdf (accessed Mar. 9, 2024).
[14] UNITQ, https://www.unitq.com/ (accessed Mar. 9, 2024).
[15] “Error and real user monitoring,” BugSnag, https://www.bugsnag.com/ (accessed Mar. 8, 2024).
[16] “Work wiser with workforce analytics & productivity insights,” ActivTrak, https://www.activtrak.com/ (accessed Mar. 8, 2024).
[17] “Audience response: Polling: Compliance voting: Q&A: Quiz: Surveys,” MeetingPulse, https://meetingpulse.net/ (accessed Mar. 8, 2024).
[18] “Audience interaction made easy,” Slido, https://www.slido.com/ (accessed Mar. 8, 2024).
[19] “Interactive presentation software,” Mentimeter, https://www.mentimeter.com/ (accessed Mar. 8, 2024).
[20] What is SDLC? - software development lifecycle explained - AWS, https://aws.amazon.com/what-is/sdlc/ (accessed Mar. 9, 2024).
[21] “Bleep-41488.MP4,” Google Drive, https://drive.google.com/file/d/103jtpbC1E-IfM_dwkqKTq0rclay7_uqy/view?usp=sharing (accessed May 4, 2024).
[22]“MediaPipe,” Google Developers. [Online]. Available: https://developers.google.com/mediapipe. [Accessed: 13-Mar-2024].
[23] “Long Short-Term Memory (LSTM), clearly explained,” YouTube, https://www.youtube.com/watch?v=YCzL96nL7j0 (accessed May 4, 2024).
[24] I. Kas, “Ilke-kas/dipproject,” GitHub, https://github.com/ilke-kas/DIPProject (accessed May 6, 2024).
[25] Ilke-Kas, “DIPProject/DatasetPreparation/labels.csv at main · Ilke-KAS/dipproject,” GitHub, https://github.com/ilke-kas/DIPProject/blob/main/DatasetPreparation/labels.csv (accessed May 6, 2024).

[26] "Videos," Google Drive, https://drive.google.com/drive/folders/1W3dgOYLHl5CmaZqupasvGZz_hVRQ7oGI?usp=sharing (accessed May 6, 2024).
[27] "Attention_Model.ipynb," Google colab, https://colab.research.google.com/drive/1CC1o9xPHbpJg8zmE7Jf_lSAsPUoacrJj (accessed May 6, 2024).

9. Appendices

Appendix A - Informed Consent Form

Informed Consent Document

Title of Project: Oculus Vigilis:Real-Time Attention Model

Investigators: Ilke Kas
PhD Student
Department of Electrical, Computer, and Systems Engineering
Case Western Reserve University
+1 (440) 773 - 2574
ixk238@case.edu

Ritika Lamba
Masters Student
Department of Computer Science
Case Western Reserve University
+1 (216) 423 - 8737
ixk238@case.edu

This informed consent form is required for you to take part in a human-computer interaction session that is part of the project named above. This project aims to help people to overcome their self-perceived attention problems such as ADHD or hyperactivity by giving a real-time feedback of the attention levels of one user using their camera records, especially during online meetings and learning processes. Personal data will be collected during the study; however, this information will be stored under a random subject number such as S1 or S2, and it will be kept in a way that makes identification very unlikely (see the description below). By signing this form, you allow us to process your stored data by law.

Your participation is voluntary, which means you can choose whether to participate or not to participate after reading the information provided in this document. This informed consent form describes the study purpose, the possible risks of participation, the financial reward (if any), the tasks that you will undertake if you decide to participate, and the data we will collect. If you are not sure about this experiment, you should ask the principal investigator or an investigator for additional information. You can withdraw from the experiment and revoke your consent at any time without any consequences, even after the end of the experiment. To do so, please contact Ilke Kas whose contact information is listed on the first page of this document.

If any part of this document is unclear to you, please ask the investigator to explain what you do not understand before signing this form. If you do decide to participate in this project, please sign this form at the indicated location on the last page. You should keep a copy for future reference.

What is the purpose of this study?
This project has one main objective:

- It aims to help people to overcome their self-perceived attention problems such as ADHD or hyperactivity by giving a real-time feedback of the attention levels of one user using their camera records, especially during online meetings and learning processes.

What are the requirements to participate in this study?
Anyone who has own personal computer with a camera can participate in this study.

How long does it take to complete this study?
This study will take 2 hours.

How many other people will be in this study?
There will be 5 participants in the study, depending on the outcome of the experimental sessions.

Where will the study take place?
The study will be conducted wherever the participant wants . The place should have enough illumination. The time and date of your participation will be determined through communication with the investigator.

What is my task?
We want to record your face under enough illumination to see your face in the recording video. We provided you with a video that will guide you through your recording. In this study, we want to measure the attention of the people in front of the camera. Due to that fact, we need different situated data from different people and we believe you are a good actor to pretend like you are in different situations! We need your 2 hours of recording without interruptions. Therefore, please try to complete this in your free time. PLEASE DO NOT USE EXTERNAL MONITOR OR CAMERA.

In the video we will provide you, there will be different tasks during different time periods. You do not have to memorize the tasks or order of them since we will guide you through the video we provided. We want you to perform the task given in the video while you record yourself. Between the tasks, there will be a beep sound that informs you the task is completed and it is time to start the next task. Here are the overview of the tasks (AGAIN DO NOT MEMORIZE THESE):

1. Start your video recorder
2. Start the video that we provide to you. Perform these task while record your video:

- Pose your head to right and look outside of the screen - 2 minutes
    - BEEP SOUND
- Pose your head to left and look outside of the screen - 2 minutes
    - BEEP SOUND
- Pose your head to up and look outside the screen - 2 minutes
    - BEEP SOUND

- Play with your phone without looking at screen - 4.5 minutes
    - BEEP SOUND
- Pretend like you are sleepy by closing your eyes - 4.5 minutes
    - BEEP SOUND
- Tilt your head to right and pretend like your attention decreases -4.5 minutes
    - BEEP SOUND
- Tilt your head to left and pretend like your attention decreases -4.5 minutes
    - BEEP SOUND
- Track the red laser dot with your eyes - 2 minutes
    - BEEP SOUND
- Pose your head to right and look outside of the screen - 2.5 minutes
    - BEEP SOUND
- Pose your head to left and look outside of the screen - 2.5 minutes
    - BEEP SOUND
- Pretend like you are sleepy by closing your eyes and yawn a lot - 4.5 minutes
    - BEEP SOUND
- Track the red laser dot with your eyes - 2 minutes
    - BEEP SOUND
- Pose your head to down and look outside the screen - 4.5 min
    - BEEP SOUND
- Watch Something that increase your attention - 1 hour
  This thing can be your favorite movie, TV series, TED Talks that focus your attention, Some Reels or posts from social media etc. But we want you to focus on the screen during that time. Please choose according to that.
    - BEEP SOUND

- Watch something that decreases your attention - 15 minutes
  This thing can be your less favorite movie, TV series, TED Talks that decrease your attention etc. But we want you to not focus on the screen during that time. Please choose according to that. Here are some advices from us:
    - The World's Most BORING video…– https://www.youtube.com/watch?v=lVrYV0odeFY - – tap that leaks water really boring 5 min
    - Bore Me To Sleep #2 - Unintelligible Math: Calculus I – https://www.youtube.com/watch?v=2KMXNg5mrM8
    - BEEP SOUND

After you finish your recording, you can upload your video to the drive folder that the investigators shared with you (Only the owner of the study Ritika and Ilke will see it).

What are the risks from this experiment?

The risks associated with participation in this study are not greater than those encountered in daily life. We try our best to avoid any problems that could arise for participants from this project.

From our perspective, there is no risk of this experiment. However, if you are not accustomed to looking at the computer screen for 2 hours, it may cause little eye strain.

You are also welcome to take a break at any time during the recording.

Can I leave the study before it ends?
The study may be stopped without your consent for the following reasons:

- The investigator or principal investigator feels it is best for your safety and/or health.
- The experimental equipment is not functioning as expected.
- You have not followed the study instructions.

You have the right to drop out of the user study at any time; simply contact Ilke Kas using the information provided on the front page of this document.

How will confidentiality be maintained and my privacy be protected?
The project team will make every effort to keep all the information we record during the study strictly confidential, as required by law. Any documents you sign, where you can be identified by name, will be kept in a locked archive in Dr. Block's department. These documents will be kept confidential. All signed documents will be destroyed when the study is over. The surveys you fill out and the electronic data we record will be marked with a unique subject identification number, but we will not keep any data that will connect this number with your name. The surveys and electronic data will be destroyed when the investigators decide that they are no longer needed for active projects.

The data recorded during the sessions will be stored on secure computers and servers. To protect your privacy, your name will be stored only on this paper consent form. The rest of the data will be stored by your unique subject identification number, and there will be no occurrence of your name in the digitally stored data. Unauthorized people will not have access to the stored data.

After we collected your camera recordings, we are going to extract some numerical features that do not unveil any necessary information except your attention during the recording. After the study ends, your recordings will be deleted from everywhere including the drive folder that you uploaded.

What will I receive after the experiment?
Participants will not be compensated for their participation.

Your rights
Within the scope of the legal possibilities, you have the right to obtain information regarding your stored data, the right to correct inaccurate data, and the right to demand the deletion of data in cases of inadmissible data storage and portability within the scope of the legal possibilities.

Signature
When you sign this document, you are agreeing to take part in this user study. If you have any questions, or if there is anything you do not understand, please ask the investigator. You will receive a copy of this informed consent form.

Printed Name

_____

Signature

_____

Place, Date

_____

Appendix B - Media Release Form

Media Release Form

Human-computer interaction research has become increasing popular. Research in this field has previously received media attention, including being featured on NowThis Future, IEEE Spectrum, The New York Times, NBC, SWR, and Late Night with Seth Meyers. Occasionally, media outlets ask us for video footage and images to appear alongside an article or presenter. We never share names or any other personal information about our participants outside the research team, but participants are recognizable in our videos because your face will not be blurred.

Would you allow us to share video and images of the main tasks of this study (attention measuring) with any media outlets that contact us requesting this? If yes, please read and sign the form below. Signing this form is completely voluntary; you may participate in the study without signing it. Please feel free to ask the experimenter if you have any questions.

I, _____, hereby give Ilke Kas, Ritika Lamba and other members of the project team at Case Western Reserve University the right and permission to copyright and/or publish, reproduce or otherwise use my face, voice, and likeness in video, photographs, written materials, and audio-visual recordings. I acknowledge and understand these materials about me will only be used for non-commercial purposes.

I understand that my image may be edited, copied, exhibited, published and/or distributed. I also understand this material may be used individually or in conjunction with other media in any medium, including without limitation to print publications, digital publications, and/or public broadcast for any lawful purpose. There is no time limit on the validity of this release, nor are there any geographic limitations on where these materials may be distributed.

I understand that my participation is voluntary and that I may, at any time, discontinue my involvement before signing this document. If I choose to discontinue participation, I will notify the principal parties of Case Western Reserve University by providing written notice.

I hereby certify that I am over eighteen years of age and am competent to contract in my own name insofar as the above is concerned. By signing this form, I acknowledge that I have completely read and fully understand the above consent and release and agree to be bound thereby. I hereby release any and all claims against any person or organization utilizing this material for marketing, educational, promotional, and/or any other lawful purpose whatsoever.

Participant Name (please print):

_____

Participant Signature:

_____

Place, Date:

Appendix C- Statistical Analysis for Relation Between Features

Covariances

Covariance measures the relationship and the dependency between two variables. It is calculated as in equation (iv).

$$cov(x,y) = \frac{\sum_{i=1}^{n}(x_i-\bar{x_i})\ (y_i-\bar{y_i})}{n-1} \text{ (iv)}$$

It is important to note that the covariance only measures the direction between two variables. The strength of their dependency or the relationship is not calculated by the covariance since the values are not scaled. Therefore, the covariance values can be so big or low and this does not show us the strength of their relation.

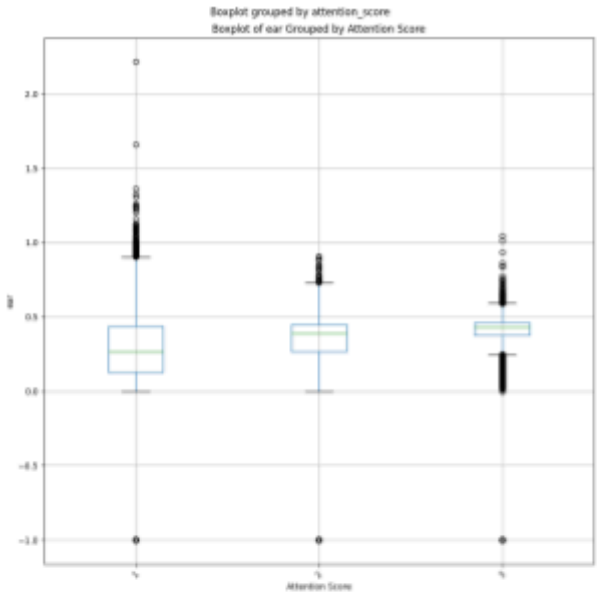
Correlations

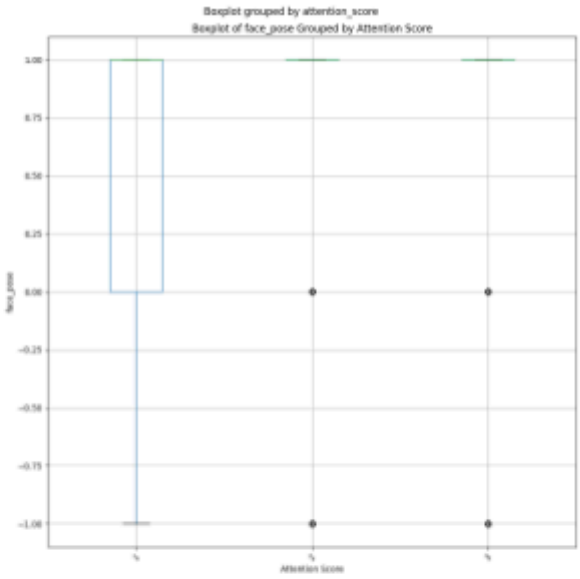
Correlation measures both the direction and the strength of the relationship and the dependency between two variables. It is calculated as in equation (v).

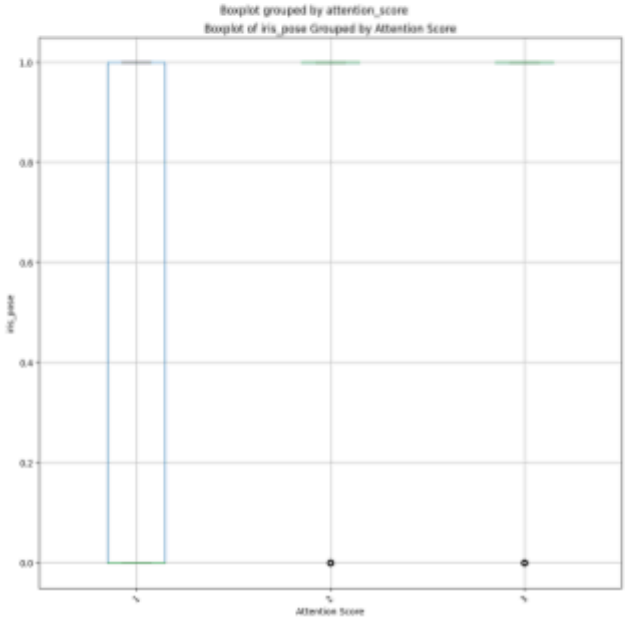
$$corr(x,y) = \frac{\sum_{i=1}^{n}(x_i-\bar{x_i})\ (y_i-\bar{y_i})}{\sqrt{\sum_{i=1}^{n}(x_i-\bar{x_i})^2(y_i-\bar{y_i})^2}} \text{ (iv)}$$

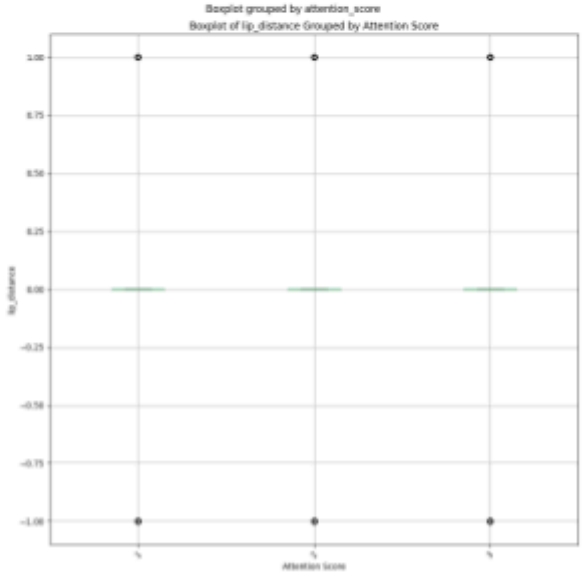
As one can notice, the correlation is scaled according to the values of the variables. Therefore, in addition to the direction of the relation between two variables, it can also give the strength of the relationship. The correlation of two variables can take values between -1 and 1. -1 shows the perfect negative linear relation between two variables while 1 shows perfect positive linear relation between variables.

Appendix D- Boxplots

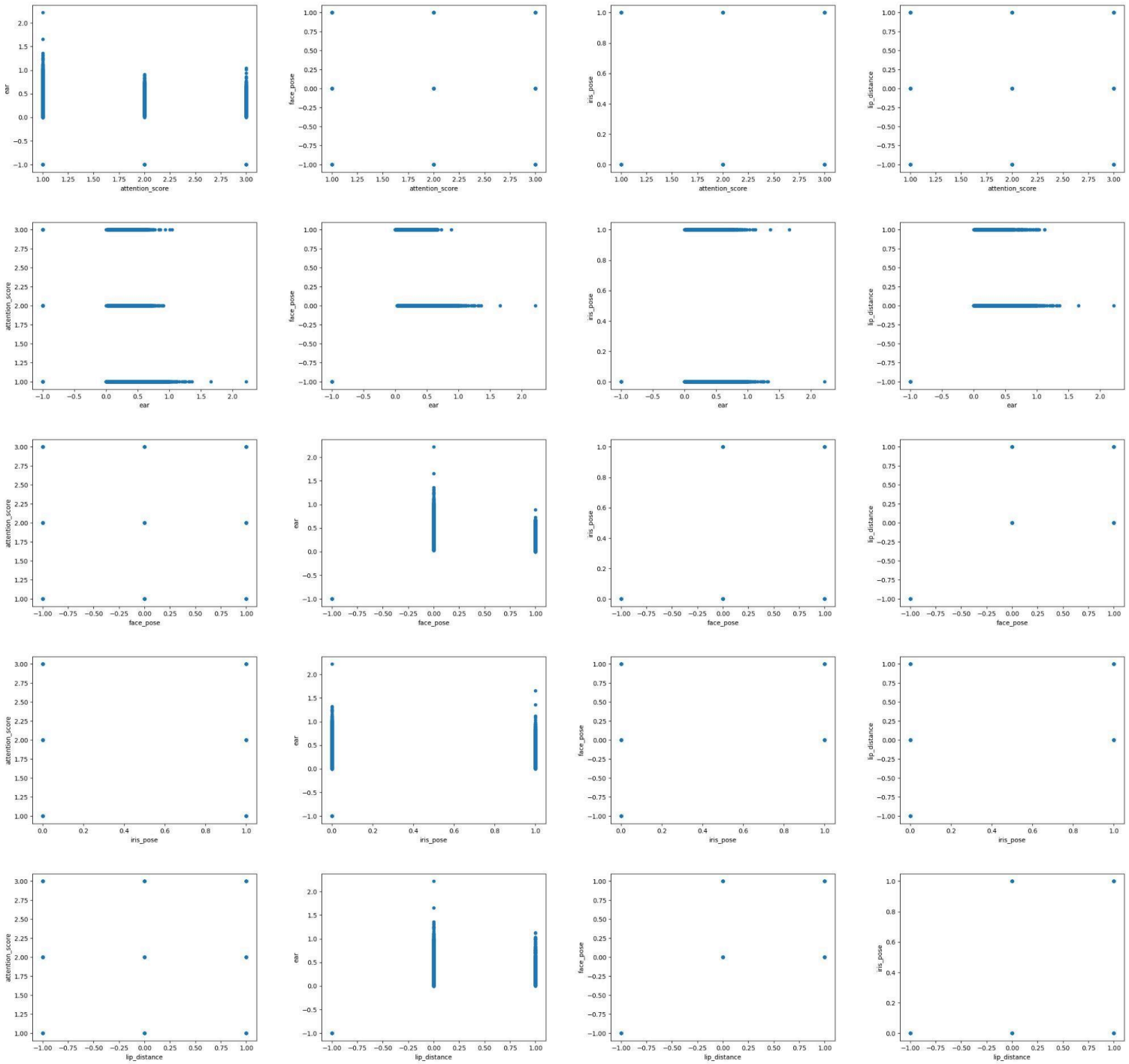








Appendix E- Scatter Plots



Appendix F- Evaluation Metrics

There are different evaluation metrics to evaluate the performance of the model. Their general rule is given below for the reader. “TP” means “True Positives”, “TN” means “True Negatives”, “FP” means “False Positives” and “FN” means “False Negatives”.

- Accuracy is the ratio of number of correct predictions to size of the dataset.

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN}$$

- Precision indicates how many positive predictions made are correct .

$$Precision = \frac{TP}{TP+FP}$$

- Recall indicates how many of the originally positive labeled data are predicted correctly by the model .

$$Recall = \frac{TP}{TP+FN}$$

- F-1 Score combines the precision and the recall. It is the harmonic mean of these two metrics.

$$F1 = 2 * \frac{Precision.Recall}{Precision+Recall}$$

The support shown in the classification reports in **Figure 16** gives the number of data whose ground truth is 0, 1 and 2. In the classification reports in **Figure 16,** one can also see that these values are calculated for each class. In addition to this, it is possible to see the macro average and weighted average calculation for metrics precision, recall and f-1. The macro average score for a metric is calculated by summing up the values of the metric for each class and dividing it to the number of classes.

- $MA_{Precision} = \frac{Precision_{score_0} + Precision_{score_1} + Precision_{score_2}}{3}$
- $MA_{Recall} = \frac{Recall_{score_0} + Recall_{score_1} + Recall_{score_2}}{3}$

- $MA_{F1} = \frac{F1_{score_0} + F1_{score_1} + F1_{score_2}}{3}$

Finally, the weighted average is calculated by multiplying the metric of a specific class with the support proportion of that class and summing up all of them.

- $WA_{Precision} = \frac{Precision_{score_0} Support_{score_0} + Precision_{score_1} * Support_{score\_1} + Precision_{score_2} * Support_{score_2}}{Support_{score_0} + Support_{score\_1} + Support_{score_2}}$
- $WA_{Recall} = \frac{Recall_{score_0} * Support_{score_0} + Recall_{score\_1} * Support_{score\_1} + Recall_{score_2} * Support_{score\_1}}{Support_{score_0} + Support_{score\_1} + Support_{score_2}}$

$WA_{F1} = \frac{F1_{score_0} * Support_{score_0} + F1_{score\_1} * Support_{score\_1} + F1_{score_2} * Support_{score_2}}{Support_{score_0} + Support_{score\_1} + Support_{score\_2}}$