SENIOR DESIGN PROJECT

Horus

Project High-Level Design Report

Ufuk Palpas, Furkan Demir, Can Kılıç, Asya Doğa Özer, Muzaffer Köksal

Supervisor: Uğur Güdükbay
Jury Members: Shervin Arashloo, Hamdi Dibeklioglu

December 24, 2021

This report is submitted to the Department of Computer Engineering of Bilkent University in partial fulfillment of the requirements of the Senior Design Project course CS491/

Table of Contents

1. Introduction **1**
- 1.1 Purpose of the System 1
- 1.2 Design goals 1
- 1.2.1 Reliability 1
- 1.2.2 User-Friendliness 2
- 1.2.3 Efficiency 2
- 1.2.4 Performance 2
- 1.2.5 Security 2
- 1.2.6 Extendibility 2
- 1.3 Overview 2

2. Current Software Architecture **3**
- 2.1 Some Current Applications-Toolkits: 3
- 2.2 Some Current Architectures and Models: 4

3. Proposed Software Architecture **5**
- 3.1 Overview 5
- 3.2 Subsystem Decomposition 5
- 3.3 Hardware/Software Mapping 5
- 3.3.1 Hardware Mapping 5
- 3.3.2 Software Mapping 6
- 3.4 Persistent Data Management 6
- 3.5 Access Control and Security 6
- 3.6 Global Software Control 6
- 3.7 Boundary conditions 6
- 3.7.1 Set-Up 6
- 3.7.2 Initialization 7
- 3.7.3 Termination 7
- 3.7.4 Failure 7

4. Subsystem Services **7**
- 4.1 Interface 7
- 4.2 Mobile Interface 9
- 4.3 Controller Component 10

5. Consideration of Various Factors in Engineering Design **11**

6. Teamwork Details **13**
- 6.1 Contributing and Functioning Effectively on the Team 13
- 6.2 Helping Creating a Collaborative and Inclusive Environment 14
- 6.3 Taking Lead Role and Sharing Leadership on the Team 15

7. References **16**

1. Introduction

1.1 Purpose of the System

The goal of Horus is to recognize human emotions. Horus will detect the emotions of a single person or a group of people. The detection can be done with a camera or multiple cameras rapidly taking images/video in real time. Horus will first detect the faces and then analyze their emotions. Horus will be able to analyze the emotions of the user from saved videos or images besides real time detection.

In the single user mode, Horus examines a saved video or real time footage over a time period, it will report the various emotions detected from a single person. So it will attempt to give the mood of the subject from long term analysis of his emotions with respect to time. For Horus to analyze a user's face, there shouldn't be any obstacles, so if the user wears a mask, hat, etc. Horus will warn the user to remove any such obstacles. Horus will also have a crowd control mode where it will analyze the emotions of a group of people such as stand-up, movie, art activity audiences, or students in a classroom through multiple cameras or a single camera to provide feedback to show organizers, who can look at audiences' moods and decide on which content to keep and which to discard. Horus will also detect emotions on video conference calls such as Zoom. For this option, Horus will contain a screen capturing mode. It will analyze the emotions of the participants on the conference call and provide feedback to the organizers. Horus will also have a deception detection mode to tell whether a sentence told by the user is a lie according to the micics and the tone.

The results will be shown in real time on the desktop app with a small window or on the mobile app with audio or visuals in various types of graphs that are selected by the user. The graphs will contain detailed information about the scan such as average, time analysis, camera statistics etc. to give an appropriate result to the user.

Horus will have many usages. For example, a street artist can place a camera nearby and get feedback from the Horus according to the emotions of the audience by using a mobile application that shows the general mode of the crowd. Another use case can be handling tension and preventing aggression. For example, Horus can be used in a prison's common areas to keep an eye on prisoners. If Horus detects multiple nervous, angry prisoners, it will alert the guards.

1.2 Design goals

1.2.1 Reliability

- Horus should detect the faces as correctly as possible.
- Horus should detect the emotions as correctly as possible.
- Horus should determine deception as much as possible.

1.2.2 User-Friendliness

- The user interface of our app should be simple and easily understandable.
- It shouldn't require any pre-education to use it.
- The user should get audible feedback easily from his/her smartphone.
- The user should change the colors and types of the resulting graphs as s/he desires.
- The user should see the real-time and cumulative results at the time of scanning easily on the same screen or from his/her phone.
- The user should be able to reach older results and view them easily.

1.2.3 Efficiency

- It should detect the general mood of the user with a long enough video.
- In crowd control mode, Horus should calculate the average emotion of the crowd.
- The results of the detection should also be given according to time.
- Regardless of the type of video conference app, Horus should start scanning with screen capture mode.

1.2.4 Performance

- Horus should use the resources (such as CPU, RAM, etc.) of the computer which it runs to give accurate results in real-time with minimal delay to give immediate feedback to the user.

1.2.5 Security

- The photos/videos that our app captured will not be shared via any third parties by Horus.

1.2.6 Extendibility

- Horus can be extended to involve different types of scans (modes) in the future.
- Horus can be modified to provide data on a certain demographic within an audience, not just the entire audience.

1.3 Overview

Horus mainly consists of 4 modes which are the single user mode, crowd control mode, deception detection mode, and the screen capture mode, and an option to view the results of older scans. The scans can be done by using input from a single camera, multiple cameras, uploaded photos/ videos, or capturing screen. The multiple camera option can only be used in the crowd control mode and uploading from a local device can be only used in crowd control and single user modes. The deception detection accepts only real time input and capturing screen mode can be chosen for only real time screen capturing.

The results of the scan are available in real time and the user can see them while making the scan. After the scan is finished the scan is saved to the local device of the user and the user can view older scans. There are multiple graph alternatives in which the user can view the results and it is possible to download the results in the desired format.
Moreover, a mobile application to track the current scan or to view older scans is available. The device can be connected with Bluetooth. Audible and visual feedback are possible within this mobile application.

2. Current Software Architecture

Lately, computer vision is developing in a considerably fast manner. This allows people to develop new technologies related to human behaviors, feelings, and emotions. Therefore, in the market, various applications contain some features of the Horus such as emotion recognition. However, Horus differs from those applications by consolidating various features of them, and by adding many ultimate features. Although Horus uses the same base as those applications, it has a unique aim and unique features. The most significant features of Horus, making the difference, are:

- The capability of emotion recognition of a group of people in real time video record and online conferences,
- Opportunity to make inferences according to results by using charts,
- Multi-camera support,
- Mobile application connection with audible warning for feedback.

2.1 Some Current Applications-Toolkits:

- **FaceReader powered by Noldus:** A desktop application to detect the facial expressions of people. Although it can generate detailed charts after the analysis of a video containing one face, it cannot detect the average mood of a group of people [1].
  FaceReader is a proprietary program; therefore, its software architecture is not completely given. However, it is known that it finds the faces using Deep Learning based face determination algorithms. Then it creates an accurate face model using approximately 500 key points and finally, it classifies the expressions with artificial neural networks.

- **Morphcast:** It is an emotional AI that can detect the gender, emotions, attention of the user. However, unlike Horus, it cannot give solutions to the user, and it is not possible to use it in different fields such as video conferences, security cameras. It can only be used for testing purposes [2]. Unfortunately, the software architecture of Morphcast is private.

- **F.A.C.E API from Sightcorp:** It is a close purpose application to Horus. It can be thought of as an all-in-one program that contains various features including some features of the Horus. However, It mostly focuses on face detection in crowded areas and measuring crowd

demographics such as calculating the age or gender of people in a group. This makes Horus different from the F.A.C.E API [3].

- **EmoPy:** EmoPy is a python toolkit containing deep neutral net classes. It can detect human emotional expression classification from a given image. It uses some public datasets and convolutional models built using the Keras framework [4].

- **Haar-cascade:** Haar-cascade is a face detection module involved in OpenCV which can be used for real-time or image data. It offers pretrained models. Those models can also be retrained or trained models can be trained more with new data. Its algorithm uses line and edge detection [5].

2.2 Some Current Architectures and Models:

Lately, many machine learning models, especially different types of convolutional neural network architectures have been trained with datasets including pictures of different kinds of human emotions. According to the paper named "Facial Emotion Recognition : State of the Art Performance on FER2013", the accuracy rate of some models had been tested. Table 1 gives the accuracy rates of various CNN and similar architectures tested in that paper. To achieve the best performance from the model different optimizers had been tried and the best learning rate had been chosen and weights were hyper-tuned.

Table 1: Neural network architectures and machine learning models used for emotion detection with the accuracy rates [4]

Method	Accuracy Rate
CNN [5]	62.44%
GoogleNet [6]	65.20%
VGG+SVM [7]	66.31%
Conv+Inception Layer [8]	66.40%
Bag of Words [9]	67.40%
Attentional ConvNet [10]	70.02%
CNN+SVM [11]	71.20%
ARM(ResNet-18) [12]	71.38%
Inception [13]	71.60%
ResNet [13]	72.40%
VGG [13]	72.70%

3. Proposed Software Architecture

3.1 Overview

The proposed software architecture will be explained by showing subsystem decomposition and clarifying every layer of it. Details of our system are described with diagrams and classes. In addition to it, software mapping, persistent data management, access control security, global software control, and boundary conditions are explained in detail. At last, the initialization, termination, and failure conditions are mentioned.

3.2 Subsystem Decomposition

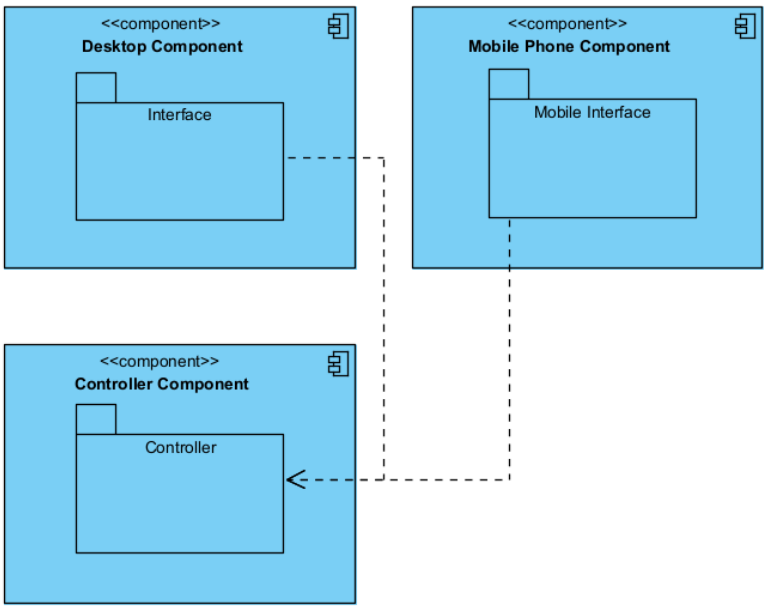

Figure 2: Subsystem Decomposition

As we were deciding on the program's design decisions, we concluded on a subsystem as given above. We divided the system into packages and components. The subsystem components are depicted in Figure 2. They are examined in detail in Section 4.

3.3 Hardware/Software Mapping

3.3.1 Hardware Mapping

Horus will not need any hardware mapping other than the cameras' mapping. While running Horus on the user's personal computer, receives data from the active cameras. Furthermore, according to the user's choice, there will be a connection between the mobile device and the user's personal computer. This communication will be held over the Bluetooth chips of the devices.

3.3.2 Software Mapping

Horus' algorithm will be implemented via Python 3 since it is suitable for computer vision projects. In addition, Python 3 contains the OpenCV library which is essential for our project. For the user interface, JavaFX will be used as it is easy to use and can be used with the Python 3 classes that will be used. In addition to these, Horus will be developed by using various datasets. Some examples of these datasets are: CrowdHuman [14] and WWW Crowd [15]. These datasets will be used for human face detection in multi-person scenes for crowd control mod. Besides these different datasets will be used inside Horus for different features.

3.4 Persistent Data Management

Horus will save the generated results in the graph format on the local hardware. It will analyze the data with graphing techniques to generate those graphs. It will also be able to save video depending on the user's choice. It will not save any data in the mobile application other than the provided graphs and charts. Furthermore, the user preferences will be stored in the local storage of the user's personal computer. The data that is not saved by the user will be deleted at the end of processing. The models that will be used for computer vision computations will also be stored in local storage.

3.5 Access Control and Security

Horus will directly connect from the mobile device to the user's local hardware, therefore access control will not be required. The user can only access the information recorded on the local storage. The recorded or processed data is only saved to the local storage otherwise, all the data will be deleted from all the systems. Horus does not require any internet connection therefore, it does not need any security constraint about other users who have access over a network connection. The local data is secured by the OS's passwords.

3.6 Global Software Control

To be event driven a project should give information to different users at the same time and the flow of the program is determined by events such as user actions, sensor inputs or message passing from other programs or threads[18]. However, Horus is an algorithm based project that will inform only one user and it will not wait for any input while working. So, Horus has no global software control.

3.7 Boundary conditions

3.7.1 Set-Up

The user will set up his/her cameras in a fashion that will allow them to view all the audiences. Alternatively, the user may start the screen recording for analysis over video

conference, or the user can select a video that he/she wants to analyze over emotions of the people recorded before.

3.7.2 Initialization

Horus can be used with a personal computer and a mobile phone which can be used to monitor results instantly. The user will launch the program and then, he/she needs to allow Horus to connect cameras, screen share or local storage to access a recorded video. Without having permission due to privacy regulations, Horus will shut itself down and will not give permission to go forward. If the user provides permission then he/she will be able to test whether faces can be detected in the current setup.

3.7.3 Termination

The program will terminate once the video/recording is ended or when the user presses the cancel button, by giving a message to the user. The program will then prompt the user if he/she wants to save the data or not.

3.7.4 Failure

Horus will stop working if all the cameras or screen share become disconnected. Horus will work as accurately as possible until all the cameras are disconnected. Also, if an unexpected power loss occurs Horus will stop working. For the cameras' disconnection, sustaining a strong connection between pc/smartphone and cameras is the users' responsibility. As to power loss, Horus will save the data during its' work time every 30 seconds to prevent the user from losing all his/her data. Thus, incase of a power cut the user will be able to recover the processed data.

4. Subsystem Services

4.1 Interface

Our outermost package is the Interface package (see Figure 3) which consists of classes that are responsible for the interface of the program. This package aims to provide a user interface that displays the results, provides the navigation, and so on.

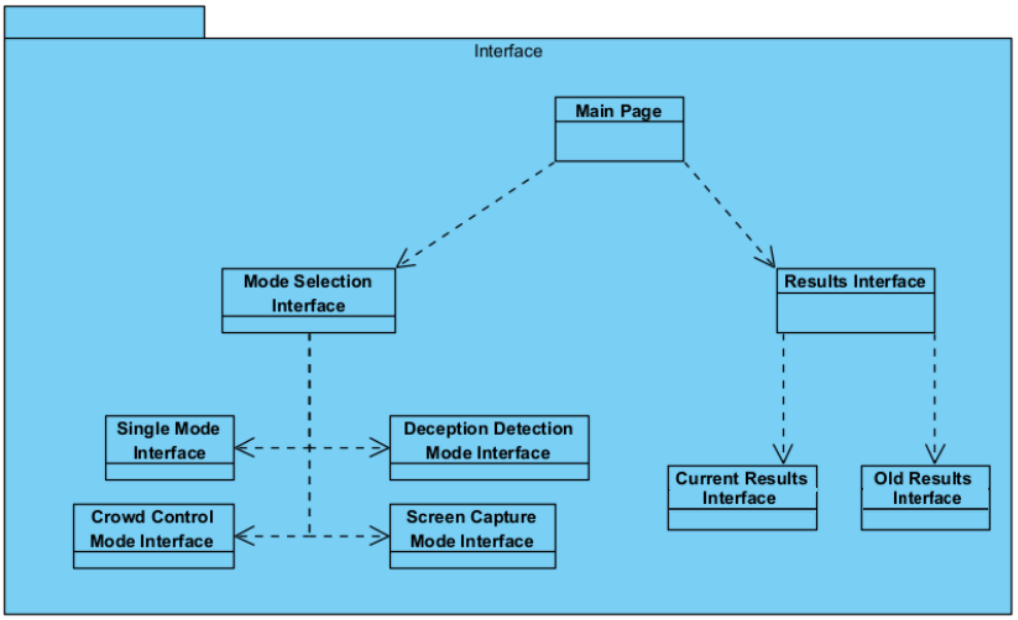

Figure 3: General Interface

- **Main Page:** Main Page is responsible for the welcoming page that has links to the given interfaces.

- **Mode Selection Interface:** The mode selection interface is responsible for handling the selection between mode options: "single user", "deception detection", "crowd control" or "screen capture".

- **Single Mode Interface:** Single mode interface is responsible for handling the single user mode functionalities.

- **Crowd Control Mode Interface:** The crowd mode interface is responsible for handling the main crowd mode functionalities and camera selection.

- **Deception Detection Mode Interface:** The deception detection mode interface is responsible for handling the deception detection mode functionalities.

- **Screen Capture Mode Interface:** The screen capture mode interface is responsible for handling the screen capture mode functionalities.

- **Results Interface:** Results interface is responsible for "choose analysis" and "choose analyses type" pages.

- **Current Results Interface:** Current results interface is responsible for showing the current results page and showing the current results inside mode pages.

- **Old Results Interface:** Old results interface is responsible for showing the old results page.

4.2 Mobile Interface

The mobile interface package (see Figure 4) contains classes that display the face and emotion detection algorithms' data in graphs and charts within the user's mobile device. This package aims to provide similar functionality to the interface package within mobile devices.

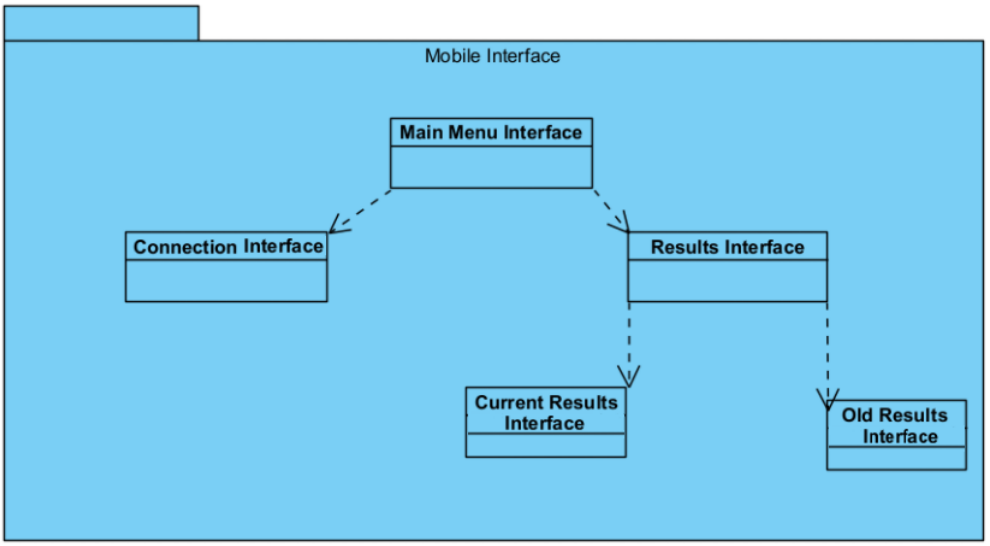

Figure 4: Mobile Interface Package

- **Main Menu Interface:** Main menu is responsible for the welcoming page that has links to the given interfaces.

- **Connection Interface:** The connection interface is responsible for showing the Bluetooth connection status.

- **Results Interface :** Results interface is responsible for "choose analysis" and "choose analyses type" pages.

- **Current Results Interface:** Current results interface is responsible for showing the current results page and showing the current results inside mode pages.

- **Old Results Interface:** Old results interface is responsible for showing the old results page.

4.3 Controller Component

The Controller package (see Figure 5) consists of classes that are responsible for data processing that comes from the Interface package. These controller classes provide emotion detection algorithms and result displaying functions.

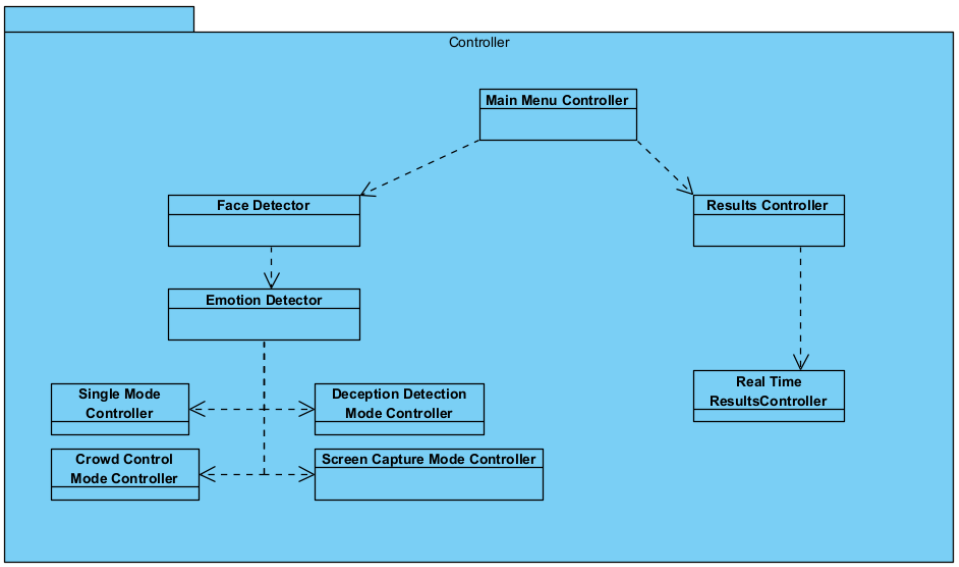

Figure 5: Controller Package

- **Main Menu Controller:** Main Menu Controller is responsible for interactions that come from the corresponding interface. Furthermore, the Main Menu Controller handles the connection operation with the mobile app.

- **Face Detector:** Face Detector detects faces within the image and frames them for the emotion detector to process. It can detect multiple faces within an image.

- **Emotion Detector:** Emotion Detector individually detects emotions of the faces provided by the face detector.

- **Single Mode Controller :** Single Mode Controller is using the Emotion Detector class for emotion detection algorithm on a single user. Then, forward the output data to be displayed on the selected type of graph.

- **Crowd Control Mode Controller:** Crowd Controller Mode Controller is using the Emotion Detector class for the emotion detection algorithm on multiple faces. It examines all the faces captured and generates an average output. Then, forward the output data to be displayed on the selected type of graph.

- **Deception Detection Mode Controller:** Deception Detection Mode Controller is using the Emotion Detector class for emotion detection algorithm on a single user and alters it to detect any kind of deception.

- **Screen Capture Mode Controller:** Screen Capture Mode Controller is using the Emotion Detector class for the emotion detection algorithm on multiple faces that are present on the screen via video conference call. It examines all the faces captured and generates an average output. Then, forward the output data to be displayed on the selected type of graph.

- **Results Controller:** Results Controller takes the data that has been generated by another controller and generates a graph of the selected type.

- **Real Time Results Controller:** Real Time Results Controller takes the data that has been generated by another controller and updates it frequently with the new data coming in from the controllers and displays the graph of the selected type.

5. Consideration of Various Factors in Engineering Design

While doing a project, consideration of various factors is really important. These various factors are public health, public safety, public welfare, cultural factors, global factors, and social factors respectively. Horus, due to its features, does not have any impact on some factors and has an impact on others. The detailed representation can be seen in Table 2.

- **Public Health:** Public health has got a few effects on the design choices of the project. For instance, mask mandates due to Covid-19 would affect the performance of Horus as it requires open faces to work properly.

- **Public Safety:** Public safety has a prominent effect on the design choices of the project. Horus' design considers whether people's privacy is encroached upon. It takes measures to not single out an individual and only provide depersonalized data.

- **Public Welfare:** Public welfare has not got a significant effect on the design choices of the project. Horus' performance is affected by the quality of the hardware devices, therefore directly related to the budget of users.

- **Cultural Factors:** Cultural factors have a prominent effect on the design choices of the project. Horus is meant for most avenues where audiences are involved; however, cultural venues such as movie theaters and stand-up shows are some of the areas where Horus will be most useful. Horus will therefore be created with such venues in mind.

- **Environmental Factors:** Environmental factors have a considerable amount of effect on the design choices of the project. Bad weather conditions may affect the performance of Horus as substances such as dust may hinder the view of the cameras in open environments. In addition, such conditions may cause damage to cameras and the local hardware if precautions are not taken against it. Light conditions may also affect the performance of Horus as it will get harder to detect faces and their emotions if it is too dark or too bright.

- **Global Factors:** Global factors have some effect on the design choices of the project. Horus aims to reach a large number of users in various countries. For this purpose, Horus' interface will be in English.

- **Social Factors:** Social factors have a crucial effect on the design choices of the project. As digital environments become more prominent in daily social interactions, the need to interpret and project real life actions and emotions into digital media becomes more important. Horus allows the user to interpret such information and provides data that can be used in digital social hubs. Therefore, it is highly relevant in this area. Horus will also consider issues of inclusivity in its modeling such as providing accurate results for different race groups.

- **Economic Factor:** Economic factors have a prominent effect on the design choices of the project. The quality and the performance of the CPUs and the GPUs will directly affect the developing and improvement processes of the Horus. The higher the performance, the less time the training process of the models will take, therefore there is a direct relation between the money spent on those devices and the developing speed.

Table 2: Factors that can affect analysis and design.

	Effect Level	Effect
Public Health	7	Masks have a negative effect on the accuracy of Horus.
Public Safety	8	Depersonalized data will be prioritized.
Public Welfare	2	Quality according to the users' budget.
Cultural Factors	7	Functionality and user experience will be geared towards cultural venues.
Global Factors	5	Horus should be English.
Social Factors	10	The need to interpret and project real life actions and emotions.
Environmental Factors	7	Light conditions and too many moving particles could affect face and emotion detection packages negatively.
Economic Factors	8	Required time for developers according to the project budget.

6. Teamwork Details

Horus is implemented and designed by five Bilkent University CS department students. All of the team members know each other from various previous projects. In other words, all of us have an idea about others' strengths and weaknesses. Because we are all close friends, we find it easier to work as a team. For every work package, we believe that we gave proper deadlines and we will be able to finish it in time. Moreover, the meetings that we hold are extremely important to make decisions about design, implementation and to have an idea on which member is in which step of his/her task.

6.1 Contributing and Functioning Effectively on the Team

We are all aware of the importance of teamwork. And this can be achieved by giving effort to the project equally. To keep track of this we will use:

- ➢ Being active on the project's Discord channel while talking about the Horus.
- ➢ The number of GitHub commits.
- ➢ The succession of given tasks.
- ➢ To keep track of the stage of the given task through the weekly meetings.

If everything goes according to plan, and if there are no serious conditions like sickness, etc. we should all have similar counts on each parameter listed above. If one or more people show less count then he/she should be motivated by the other team members because we do not have a relationship between the boss and his/her employees. Instead, we are all equal and important members of this team. However, to achieve good management over the determined tasks, we assigned a person in charge for each task set such as UI or face detection. Thus, we achieved equal workload as both responsible and workers.

To achieve equal workload tasks are planned according to full-stack development. To provide stability over the progress of tasks, we arrange evenly spaced meetings. In these meetings, we talk about what we did after the last meeting, what we will do until the next meeting. We also discuss newly or previously thought ideas, and we determine the next tasks and assign them to the group members. We generally make the task sharing according to our previous experiments and volunteering. We also will arrange spontaneous meetings if needed in cases such as quick questions of confusion about a particular task.

We will maintain the communication through the Discord app via voice communication and screen share functionality. To function effectively most of the time instead of working alone we will work under a voice channel together both to motivate each other and to achieve synchronous communication. Synchronous communication will allow us to avoid waste of time and to minimize the mistakes due to communication errors. By doing this we will be able to work a lot more effectively.

6.2 Helping Creating a Collaborative and Inclusive Environment

To create a collaborative and inclusive environment first, a big task should be divided into smaller multiple tasks. So, every team member can take a task according to a plan and give effort to it. This must be done to avoid giving some members a huge workload while giving others less. This way each member can give all their focus to a single or multiple assigned tasks. If every team member can satisfy this, then we can be sure that Horus will achieve good results and will be completed in time without any problems. Also, merging these smaller tasks should be done carefully and every team member should test whether the tasks they did fit the bigger task nicely and smoothly. Then, those smaller tasks forming a big task must also be checked by the team leader of that work package to avoid problems in merging. This double checking will also minimize the mistakes that can occur.

All of the group members have one or two main tasks and many small tasks. As aforementioned, we shared responsibility roles of the main tasks according to the requirements of the task and previous experiences of the assigned one. On the other hand, we solved and planned to solve the design-related things, and crucial project decisions together involving opinions of each member.

6.3 Taking Lead Role and Sharing Leadership on the Team

Being under the management of a leader is a crucial factor in a team work to avoid misunderstandings and confusion. Moreover it helps the team to be in a more connected and organized state. To avoid overloading all the management jobs to one single person, we preferred to divide the work into teams and assign one leader to each team, thus the workload is divided into members equally.

To share the leadership, we have seven work packages: Face Detection, Emotion Detection, Deception Detection, User Interface, Reports, Mobile Application, and Testing, respectively. We assigned those work packages to different team members. The team members with work packages with lower workloads are given a second work package with again a lower work package resulting. This way some team members got one big work package while some got two work packages smaller than the others. The leaders and members of these work packages can be seen in Table 3.

Table 3: Work Packages and responsibilities of the team members

WP#	WP title	Leader	Members
WP1	Face Detection	Asya Doğa Özer	Asya Doğa Özer, Ufuk Palpas
WP2	Emotion Detection	Ufuk Palpas	Ufuk Palpas, Furkan Demir, Muzaffer Köksal
WP3	Deception Detection	Furkan Demir	Furkan Demir, Muzaffer Köksal
WP4	User Interface	Asya Doğa Özer	Asya Doğa Özer, Can Kılıç
WP5	Reports	Can Kılıç	Can Kılıç, Asya Doğa Özer, Furkan Demir, Ufuk Palpas, Muzaffer Köksal
WP6	Mobile Application	Muzaffer Köksal	Muzaffer Köksal, Can Kılıç
WP7	Testing	Can Kılıç	Can Kılıç, Furkan Demir

7. References

[1] “Facial expression recognition software,” *Noldus.com*. [Online]. Available: https://www.noldus.com/facereader. [Accessed: 14-Nov-2021].

[2] “Facial expression and emotion recognition AI,” *Morphcast.com*, 01-Apr-2019. [Online]. Available: https://www.morphcast.com. [Accessed: 02-Nov-2021].

[3] Sightcorp, “Sightcorp,” *Sightcorp.com*. [Online]. Available: https://face-api.sightcorp.com. [Accessed: 02-Nov-2021].

[4] *EmoPy: A deep neural net toolkit for emotion analysis via Facial Expression Recognition (FER)*.

[5] “OpenCV: Cascade Classifier,” *Opencv.org*. [Online]. Available: https://docs.opencv.org/3.4/db/d28/tutorial_cascade_classifier.html. [Accessed: 23-Dec-2021].

[6] Y. Khaireddin and Z. Chen, “Facial emotion recognition: State of the art performance on FER2013,” *Arxiv.org*. [Online]. Available: https://arxiv.org/pdf/2105.03588.pdf.

[7] K. Liu, M. Zhang, and Z. Pan, “Facial Expression Recognition with CNN Ensemble,” in Proceedings - 2016 International Conference on Cyberworlds, CW 2016, 2016.

[8] P. Giannopoulos, I. Perikos, and I. Hatzilygeroudis, “Deep learning approaches for facial emotion recognition: A case study on FER-2013,” in Smart Innovation, Systems and Technologies, 2018, vol. 85, DOI: 10.1007/978-3-319-66790-4_1.

[9] M. I. Georgescu, R. T. Ionescu, and M. Popescu, “Local learning with deep and handcrafted features for facial expression recognition,” IEEE Access, vol. 7, 2019, DOI: 10.1109/ACCESS.2019.2917266.

[10] A. Mollahosseini, D. Chan, and M. H. Mahoor, “Going deeper in facial expression recognition using deep neural networks,” in 2016 IEEE Winter Conference on Applications of Computer Vision, WACV 2016, 2016, DOI: 10.1109/WACV.2016.7477450.

[11] R. T. Ionescu, M. Popescu, and C. Grozea, “Local Learning to Improve Bag of Visual Words Model for Facial Expression Recognition,” Work. challenges Represent. Learn. ICML, 2013.

[12] S. Minaee and A. Abdolrashidi, “Deep-emotion: facial expression recognition using attentional convolutional network,” arXiv. 2019, DOI: 10.3390/s21093046.

[13] Y. Tang, “Deep learning using linear support vector machines,” arXiv, vol. 1306.0239, 2013.

[14] S. Z. Shi, Jiawei, “Learning to Amend Facial Expression Representation via De-albino and Affinity,” arXiv, vol. 2103.10189, 2021.

[15] M. K. Pramerdorfer, Christopher, “Facial expression recognition using convolutional neural networks: state of the art,” arXiv, vol. 1612.02903, 2016

[16] “Crowdhuman,” *CrowdHuman Dataset*. [Online]. Available: https://www.crowdhuman.org/. [Accessed: 23-Dec-2021].

[17] J. Shao, K. Kang, C. C. Loy, and X. Wang, “Deeply learned attributes for crowded scene understanding,” *Project Page of Deeply Learned Attributes for Crowded Scene Understanding*. [Online]. Available: https://amandajshao.github.io/projects/WWWCrowdDataset.html. [Accessed: 23-Dec-2021].

[18] “Event-driven programming,” *Wikipedia*, 02-Dec-2021. [Online]. Available: https://en.wikipedia.org/wiki/Event-driven_programming. [Accessed: 24-Dec-2021].