Bilkent University

Department of Computer Engineering

Senior Design Project

Project short-name: EyeContact

Final Report

Nazlı Abaz - 21400231
Melisa Onaran - 21301232
Sarp Saatçıoğlu - 21400375
Yunus Ölez - 21401539

Supervisor: Hamdi Dibeklioğlu

Jury Members:Çiğdem Gündüz Demir, Selim Aksoy

Innovation Expert: Çağla Çığ Karaman

Website: eyecontact.in

May 03, 2018

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 CS491/2.

Table of Contents

1. Introduction

Improvement of every society depends on the contribution of individuals in it. More people contributing to it with higher rates mean faster and bigger development.Governments endeavor to create better standards to increase production of individuals. However, individuals with disabilities do not have the same living conditions since they have special needs and thus, they cannot be as productive as others.

The aim of this project is to increase the living conditions of people who are visually impaired, and thus integrate the visually impaired into the society. These individuals cannot get proper visual feedback from the environment which causes them to have a limited communication with society. The main premise of EyeContact is that, it offers a solution to video calling (which visually impaired people have a hard time using through applications such as Skype, Google Hangouts, etc.) for visually handicapped individuals by verbalizing the visual cues given by the participator of the video call to the visually impaired. Therefore, these people are able to communicate, create and produce more.

The system is able to recognize the face of the participator and detects the facial landmarks through the video captured by the camera of the participator's computer during the video call. Through the detected landmarks, the system processes the facial expressions and returns the emotions according to their density. Then, the visually impaired person is notified through an audial feedback which is given by the visually impaired person's headphones or speakers. Also, the system processes the gaze and the shirt color of the participator additionaly to the facial expressions. This provides the visually impaired person to be more interactive in the conversation in terms of non-verbal communication.

Camera that the system uses is participator's camera. Verbal notifications is output from the platform with respect to user's choice of output method (e.g. headphone, speaker, bluetooth devices etc.). This system is based on computer vision technology in order to recognize facial expressions, the visual input taken from the camera is sent to the DigitalOcean platform so as to be processed and the verbal output is sent to the participator's platform to be presented.

This system is available for cross-platform as a web application to be used by the users. Development of the user application is done using JavaScript. Processing part and WebRTC(Real Time Communication) part is handled on DigitalOcean. For facial expression detection OpenFace is used which is based on OpenCV as its computer vision tool. DigitalOcean computing is handled with REST service and WebRTC signaling methodology in order to be efficient.

In conclusion, EyeContact is a web based application that aims to help visually impaired people during video calls. The designed application converts visual information, e.g. facial expressions and gaze of the other person in the call, to voice indicators for the visually impaired user. EyeContact allows its users to make video calls with their contacts through Google account authentication. The application analyzes and interprets the video feed during the call using computer vision techniques, and generates voice indicators that depict visual details and behavioral cues about the other party in the call.

2. Final Architecture Design

2.1 Subsystem Decomposition

For the main system of our project, we decided to have the client-server architecture style which is designed to be more general type of a 3-tier architectural style. Client side deals with only client side of the application whereas Server side is responsible for handling Database Management part of the system.

Presentation Layer is designed to include only User Interface Subsystem. This subsystem is designed to serve as interface to the user while interacting with the layer down below to make sure that there is maintainability.

Application Layer is designed to be the layer where application logic is established. In this layer, main functionalities of the system is performed such as visual-audial feedback, image processing, etc. This layer is composed of User Management Subsystem, Image Processing Subsystem, Visual-Audio Feedback Subsystem and Audio Feedback Subsystem. User Management Subsystem is responsible from actions of users and interpret its commands. Image Processing subsystem handles the image processing, afterwards Visual-Audio subsystem is be responsible for visual feedback to audio feedback transition and the Audio Feedback Subsystem translates the detection output to the audio output for it to be translated through the headphones.

Data Storage Layer consists of Data Management Subsystem which stores the user data as well as contents of the user. This subsystem provides services to upper layer subsystem which are Application Layer and Presentation Layer.

Data Processing Layer swaps in data from Data Storage Layer and the data is prepared and the computed data is passed to the Data Storage Layer.

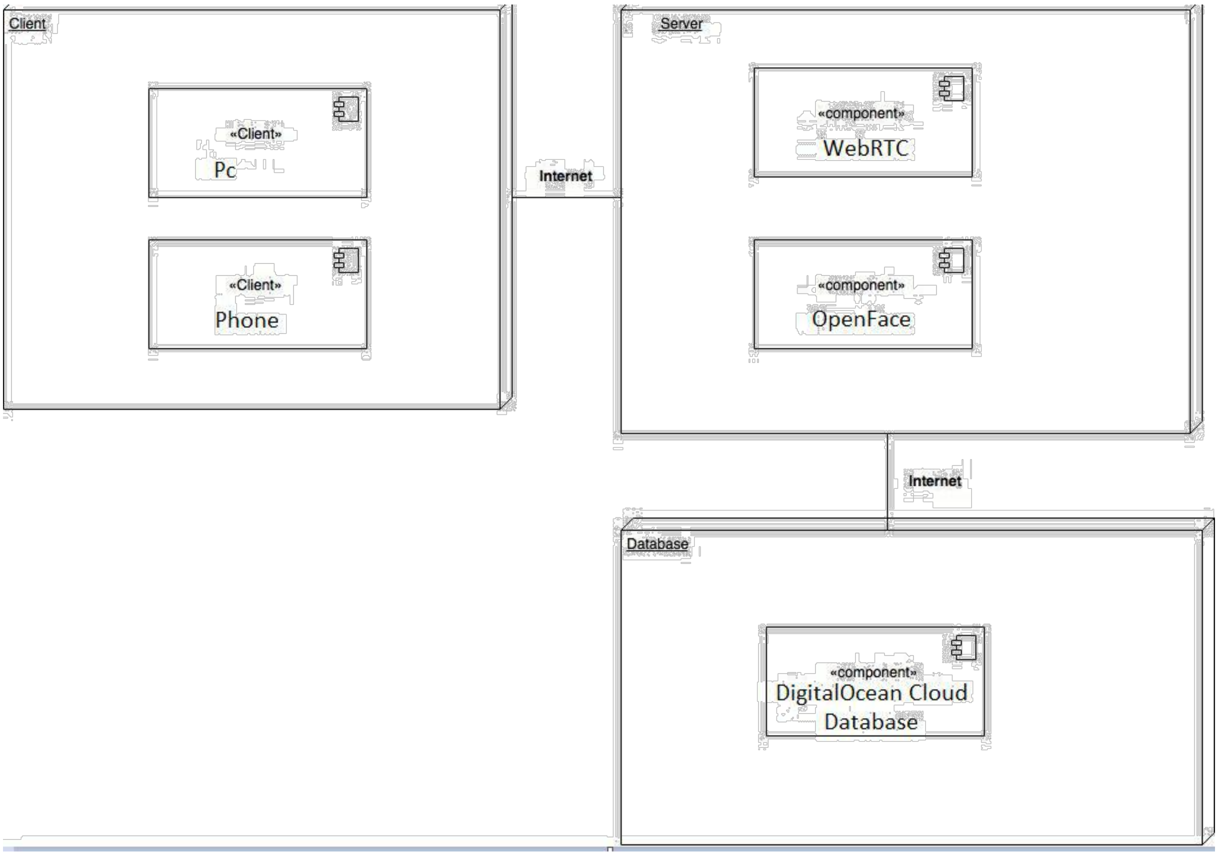

Figure 1: *Subsystem Decomposition Diagram*

2.2 Hardware /Software Mapping

User interacts with the application but to do that an internet connection is needed. Our server and database is up and running constantly so anytime users is able to use the application without any information loss or anything likewise. However as we stated above to application to work and show the relevant information or enabling its features users must have an internet connection as long as they use the application. Also WebRTC and OpenFace services must be on to be able to use the application with all of its features.

2.3 Persistent Data Management

What we aimed is to keep track of user information and the user calls constantly in our database. But since, we are already dealing with the video call and the face detection situation on a cloud server to have the best performance as possible, we thought it would be better if we ask the user to sign up with their google+ accounts, or accounts that are already memorized in their computer that they use constantly.

3. Impact of Engineering Solutions Developed in the Project

3.1 Global Impact

Currently, there are many video call applications that can be used such as Skype, Facebook Messenger, Whatsapp Video Call, Facetime, etc. However, these applications are not supported for the visually impaired people therefore it can be said that the EyeContact web application is adding a new feature to the era. Beside having a new feature compared to those applications, EyeContact is compatible with almost all of the features that the existing applications can offer. EyeContact is a video calling web application that is designed using WebRTC for its video call purpose and OpenFace for giving verbal feedback. EyeContact requires a computer with a camera or a computer and an additional camera that is connected to the computer. Application keeps the data by defining the facial landmarks to process the verbal feedback. The interface is designed to make sure that its user friendly for mainly visually impaired. Besides the verbal feedback, user can give voice command to find contact and to make the call.

3.2 Societal Impact

Nowadays, it is easy to see people with their laptops or smart phones. Despite the quality, it is quite easy to find a smart phone or a laptop with a camera that is budget proof. Nearly almost everyone has a Facebook account, uses Whatsapp and prefer Skype if none of the two. People quite often use the video call feature of this applications. In other words, having a video call is not a privilege. It only requires a smart phone or a computer with a camera. Therefore, Eyecontact is quite achievable for the society with the product they already have.

3.3 Economic Impact

One of the most important features of EyeContact is that it only requires a good internet connection, good internet connection is necessary as in every video call applications to receive qualified display. Therefore, it is a budget proof application that everyone can use.

4. Contemporary Issues Related with the Area of the Project

In video chat part of the project, the common issue is latency of video and audio due to the internet connections that are not fast enough. This latency problems are mostly solved nowadays by the power of Blob transfers through WebSocket. In EyeContact, since we use an open-source media service (Kurento) that handles the video communicatioın with Blob transfers, we almost solved this issue supposing that we have a quality internet connectiobn.

In computer vision part, most common issue is the reliability of results. We tried to increase the reliability of outputs by using OpenFace and a Native Bayesian Classifier. OpenFace is an external module written by Tadas Baltrusaitis who is a researcher working in computer vision and machine learning. The module is capable of tracking people faces and gazes. It can also output action units (AUs) on single images and live stream videos. Some different combinations of action units and their densities may represent emotions, however, OpenFace is not trained to capture emotions and for our project we need to recognize them in order to give verbal feedback to visually impaired.

There are no specific combinations or densities of AUs to define emotions and as a result, we needed to train OpenFace. In order to do so, we have used a simple machine learning technique called Naïve Bayes Classifier. Our data set to train the program contained 7 different emotions: anger, disgust, fear, happiness, sadness, surprise and neutral face. It also contained a total of 431 samples. We used 80% of each emotion's sample to train and 20% to test the classifier. Here are the numbers for each emotion group's training and testing sample size:

	Anger	Disgust	Fear	Happiness	Sadness	Surprise	Neutral	Total
Training	36	47	20	55	22	66	98	344
Testing	9	12	5	14	6	16	25	87
Total	45	59	25	69	28	82	123	431

It is known that Naïve Bayes Classifier is a really simple machine learning algorithm. Support Vector Machines (SVMs) are another machine learning technique which is a rather complex but more effective than Naïve Bayes. We thought about using SVMs, however, after analyzing test results of Naïve Bayes Classifier, in order to save time and effort we continued with Naïve Bayes. Here are the test results of Naïve Bayes Classifier:

	Anger	Disgust	Fear	Happiness	Sadness	Surprise	Neutral
Success	2	11	2	14	6	16	16
Fail	6 Sadness 1 Disgust	1 Neutral	3 Surprise	-	-	-	7 Sadness 1 Surprise
Total	9	12	5	14	6	16	25
Success Rate	22.222%	91.6667%	40%	100%	100%	100%	76%

Although these numbers seem very promising, one should consider that our testing set was small. Moreover, for instance at first sight, results of sadness appear as very succeeding. However, 6 out of 9 anger emotions and 7 out of 25 neutral emotions result as sadness. In other words, only 6 out of 19 sadness results are really correct, which decreases the success rate of sadness to 31.57%. After similar calculations done for other emotions, success rates result as follows:

	Anger	Disgust	Fear	Happiness	Sadness	Surprise	Neutral	Average
S. Rate	22.22%	84.61%	40%	100%	31.57%	80%	61.53%	59.99%

To conclude, it seems that Naïve Bayes Classifier distinguishes disgust, happiness and surprise emotions almost perfectly. It can recognize neutral faces more than half of the time; however, it cannot differentiate anger, fear and sadness emotions very well. Overall, it has approximately 60% success rate and this is enough for out project.

5. Tools and Technologies

WebRTC:

WebRTC (“Web Real-Time Communication”) is a collection of communications protocols and application programming interfaces that enable real-time communication over peer-to-peer connections.

OpenCV:

OpenCV (Open Source Computer Vision) is a library of programming functions mainly aimed at real-time computer vision.

Git/GitHub:

Git is a version control system for tracking changes in computer files and coordinating work on those files among multiple people. It is primarily used for source code management in software development. GitHub is a web-based hosting service for version control using git. It is mostly used for computer code. It offers all of the distributed version control and source code management (SCM) functionality of Git.

DigitalOcean:

DigitalOcean is a cloud infrastructure provider that provides developers cloud services that help to deploy and scale applications that run simultaneously on multiple computers.

Node.js/npm:

Node.js is an open-source, cross-platform JavaScript run-time environment that executes JavaScript code server-side. npm is a package manager for the JavaScript programming language. It is the default package manager for the JavaScript runtime environment Node.js. It consists of a command line client, also called npm, and an online database of public and paid-for private packages, called the npm registry. The registry is accessed via the client, and the available packages can be browsed and searched via the npm website.

HTTPS:

HTTP Secure (HTTPS) is an extension of the Hypertext Transfer Protocol (HTTP) for secure communication over a computer network. In HTTPS, the communication protocol is encrypted by Transport Layer Security (TLS), or formerly, its predecessor, Secure Sockets Layer (SSL).

WebSocket:

WebSocket is a computer communications protocol, providing full-duplex communication channels over a single TCP connection. The WebSocket protocol enables interaction between a web client (such as a browser) and a web server with lower overheads, facilitating real-time data transfer from and to the server.

MongoDB:

MongoDB is a free and open-source cross-platform document-oriented database program. Classified as a NoSQL database program, MongoDB uses JSON-like documents with schemas.

6. Resources

OpenFace:

OpenFace is an open-source state of the art tool intended for facial landmark detection, head pose estimation, facial action unit recognition, and eye-gaze estimation.

Kurento:

Kurento is a WebRTC media server and a set of client APIs making simple the development of advanced video applications for WWW and smartphone platforms.

Mongoose:

Mongoose is a MongoDB object modeling tool designed to work in an asynchronous environment for node.js applications.

7. References

[1] "CS491 Senior Design Project I", Ccs.bilkent.edu.tr, 2017. [Online]. Available: http://www.cs.bilkent.edu.tr/CS491-2/CS491.html. [2] "WebRTC", Wikipedia, 2017. [Online]. Available: http://www.wikizero.org/index.php?q=aHR0cHM6Ly9lbi53aWtpcGVkaW Eub3JnL3dpa2k vV2ViUlRD.

8. Appendix : User Manual

8.1 Login Page

In this page, users is asked to login to the system via his gmail account and asked for permission to allow his contacts to be copied to the system for contacts.

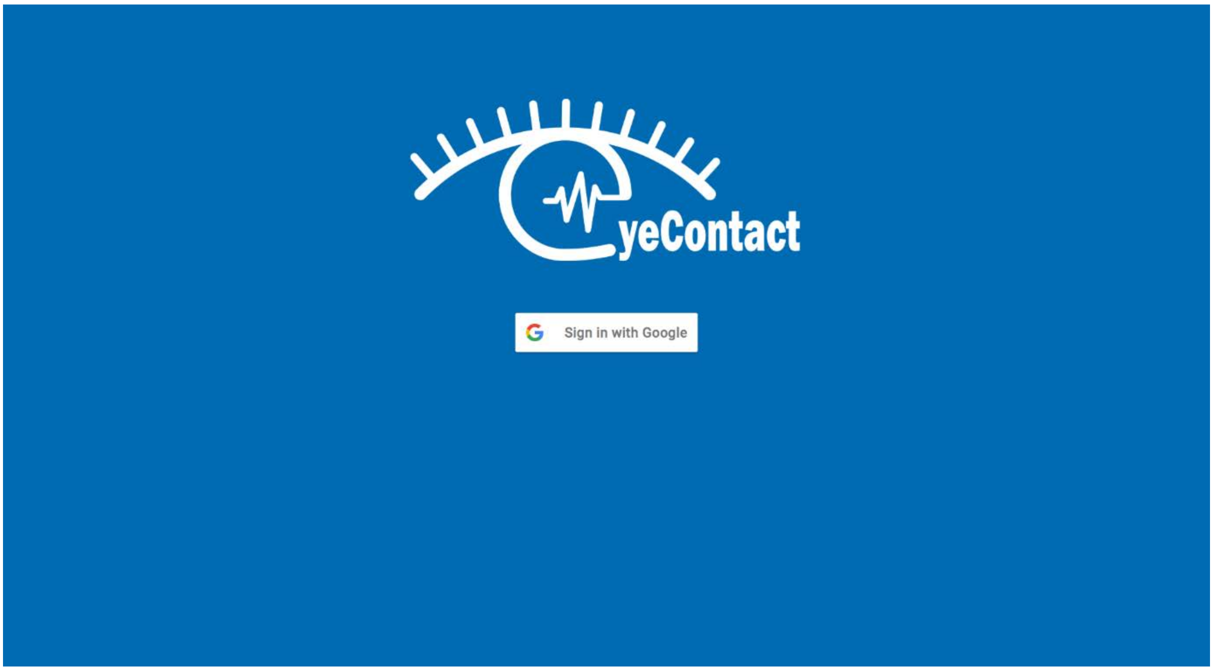

Figure 2: *Login Page*

8.2. Search Page

In that page, system guides the user with a signal voice for first to command by voice to say the contact name to the microphone, then the system lists the contacts with that name. (System works as with voice command such as Siri, system is designed that way because it is better for user experience.) Later, system ask for the user to voice command one more time with a signal voice and asks for a specific contact from the list that system made through commanded name. After the match system asks for the callee to accept or decline the call. If system cannot match a name or if the name is offline system gives verbal feedback about it and the whole process starts again.

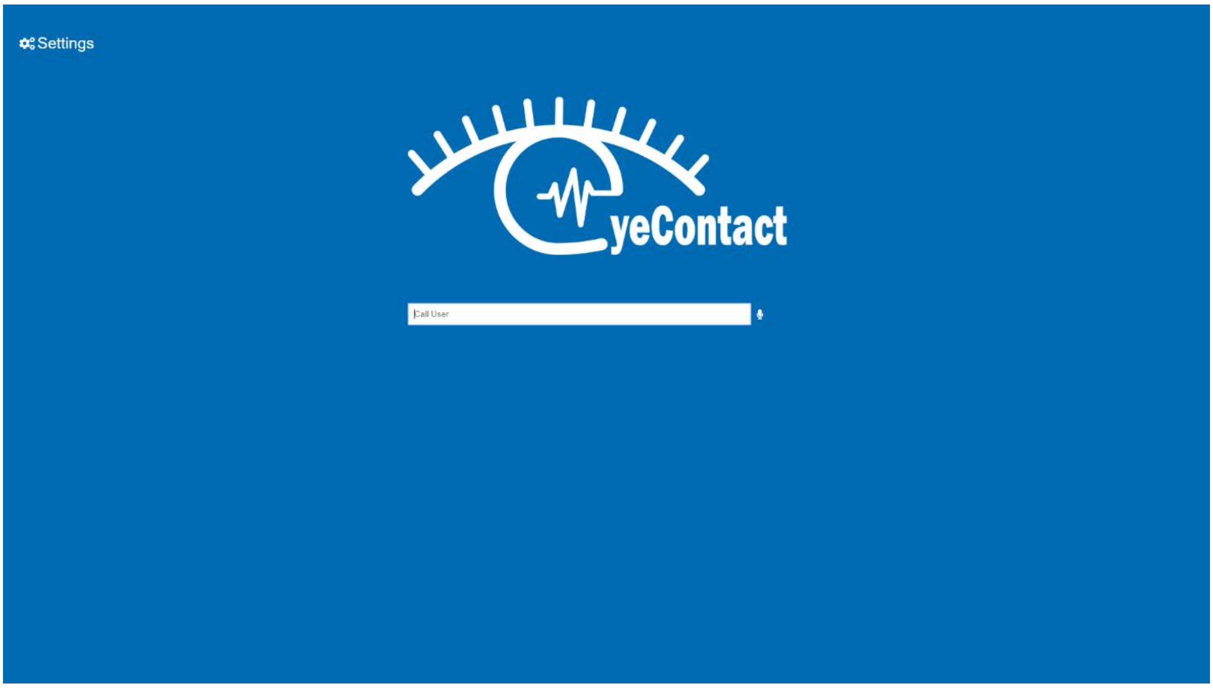

Figure 3: *Search Page*

8.3 Video Call Page

This page is where the video call happens, in this page there is a line chart and that shows the records of the emotions per time and there is another chart that show the gaze if looked left or right. Apart from that most importantly, there is a verbal feedback system that detects the facial landmarks of the callee to process and give feeback to the caller and chat can be ended with a stop button by caller or the callee and it returns to the search page.

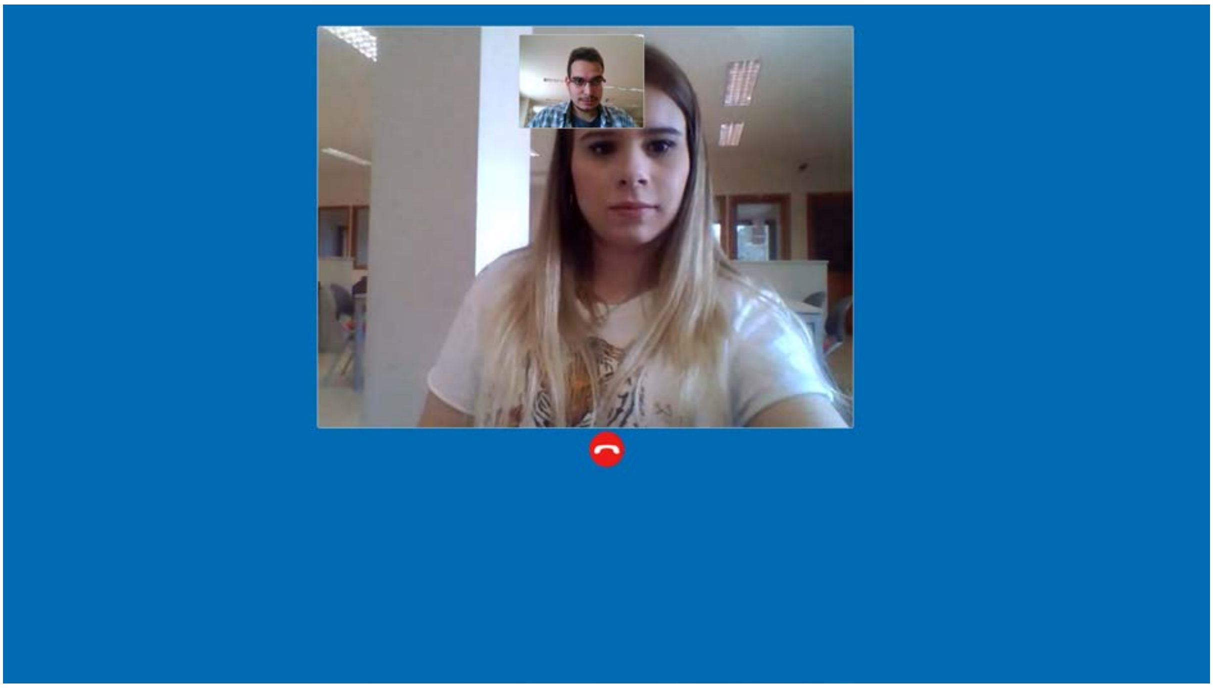

Figure 4: *Video Call Page*

8.4 Settings Page

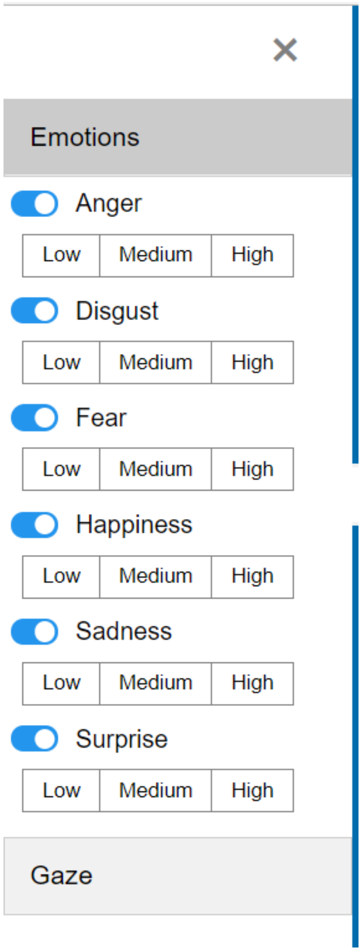

Figure 5: *Settings Page*