***Sentiment analysis using Amazon Web Services and Artificial Intelligence / Machine Learning***

***An emotional check for individuals and mental health tapping solution for preventing suicide cases***

*Charu Anant Rajput*

School of Computer Science and Engineering

VIT Chennai

Chennai, India

charuanant.rajput2019@vitstudent.ac.in

*Kunal Kumar Jha*

School of Computer Science and Engineering

VIT Chennai

Chennai, India

kunalkumar.jha2019@vitstudent.ac.in

*Bhavesh Kumar V D*

School of Computer Science and Engineering

VIT Chennai

Chennai, India

bhaveshkumar.vd2019@vitstudent.ac.in

***Abstract***

Depression and other mental ailments cause a lot of damage to an individual so we have devised this approach that can detect the mood with the help of emotion analysis based on two parameters based on facial expression and text based analysis. The face recognition classifies detected sentiment into seven categories namely happy, surprised, neutral, disgusted, angry, fearful and angry. The text based analysis classifies the detected sentiment as positive, negative, neutral and mixed. Facial expression analysis majorly makes use of face-api whereas text based analysis is using natural language processing as it’s main working principle which in the cloud environment is achieved using the AWS Comprehend service. The outcomes from both the above mentioned features can then be combined and plotted as a dynamic pie chart using google charts api. The data thus obtained from the above two features can again be used in framing a sentiment score scale that can in turn be used for predicting an individual’s sentiment in the near future.

***Keywords—AWS; Sentiment Analysis; Textual sentiments; Facial Sentiments; Data Visualization; Sentiment Score Computing Algorithm; Predictive models***

# Introduction

Depression is a very serious issue unlike physical wounds and problems. Many people don’t seek assistance or help. Mental illnesses are on peak since lockdowns are given to the issues everyone is facing and this lockdown is just a tipping point for all those who have lost their loved ones and have been living a stressful life .So we made a project to find or detect the early signs of depression in people before it ruins them.In this we’ve made a combination of face analyser and text analyser which tells your mood by reading/scanning your face image and by analysing the text you’ve written, this can be of good use for those battling depression and other mental ailments.

Also besides the pandemic situation, most of the people tend to stay away from their families due to work or education purposes.This approach can also be used as a self assessment emotional calibrator.

# Ease of Use

## The approach

The approach intends to give a precise solution for sentiment analysis and therefore decided to take multiple parameters into consideration. Face and Text analysis are the two parameters considered for developing and generating the final condition of one’s mental state. The data is available on the cloud platform where the user can have a look at the sentiment score that has been recently generated and the added advantage is that this sentiment score can be remotely accessed and generated anytime.

The solution as a whole can be divided into three broad modules. The textual emotion recognition, the facial emotion recognition and the analysis part.

(i) **Textual Emotion Recognition :** This module primarily makes use of the Amazon Web Services cloud platform and it’s services to get the input text file and display the corresponding sentiment score.

(ii)**Facial Emotion Recognition :** This module makes use of the face-api.js which is an open source facial sentiment analysis API which helps to get the sentiments of the face in real time.

(iii)**Analysis Part:** This module takes care that data from both the above 2 modules has been taken and proper scores have been obtained and stored on the cloud.

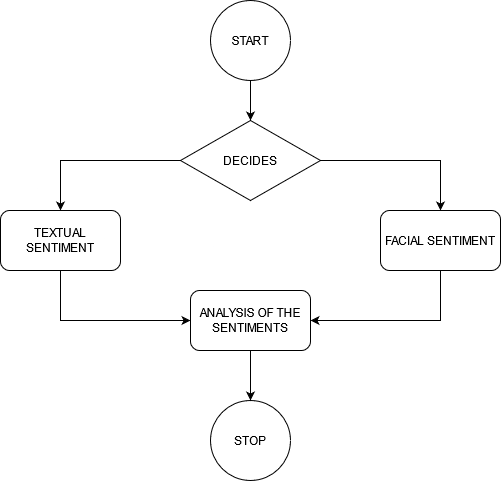
## Abbreviations and Acronyms

**AWS** Amazon Web Services ; **S3** Simple Storage Service; **IAM** Identity and Access Management; **API** Application program Interface ; **JSON** JavaScript Object Notation ;  **CSP** Content Security Policy ; **CORS** Cross Origin Resource Sharing;**XSS** Cross Site Scripting

## *Maintaining the Integrity of the Specifications*

This platform only uses the public ID of the user to identify the user and store his data accordingly.

# Technical Implementation



**Figure 1** : Technical Implementation flow chart

The user is to deal with three interfaces namely textual emotion recognition, facial emotion recognition and the final report generation. The technical implementation of each of these interfaces has been described as follows :

## Textual Emotion Recognition

* Setting up the AWS Cloud environment :

The main driving tool powering the implementation of textual emotion analysis is AWS Comprehend. Amazon Comprehend is a natural language processing (NLP) service that uses machine learning to find insights and relationships in text. It precisely analyzes text using tokenization and parts of speech.

The input text file to S3 bucket is handled by the NodeJS server using ‘AWS-SDK’ dependency which uses the S3 bucket's credentials and sends the text file directly to the S3 bucket. But the user only needs to provide the text in the textarea on the front end and NodeJS on the backend will handle the package of the text data by converting it into a file and then send it to the S3 bucket.

In order to get the necessary credentials we are to make a new IAM user using the IAM console window in order to obtain the Access Key and the Secret Key for Programmatic Access.

Next task is to create a S3 bucket which is publicly accessible. For achieving the status of public access we need to change the bucket policy and also settings have to be made to mitigate the problem of CORS.

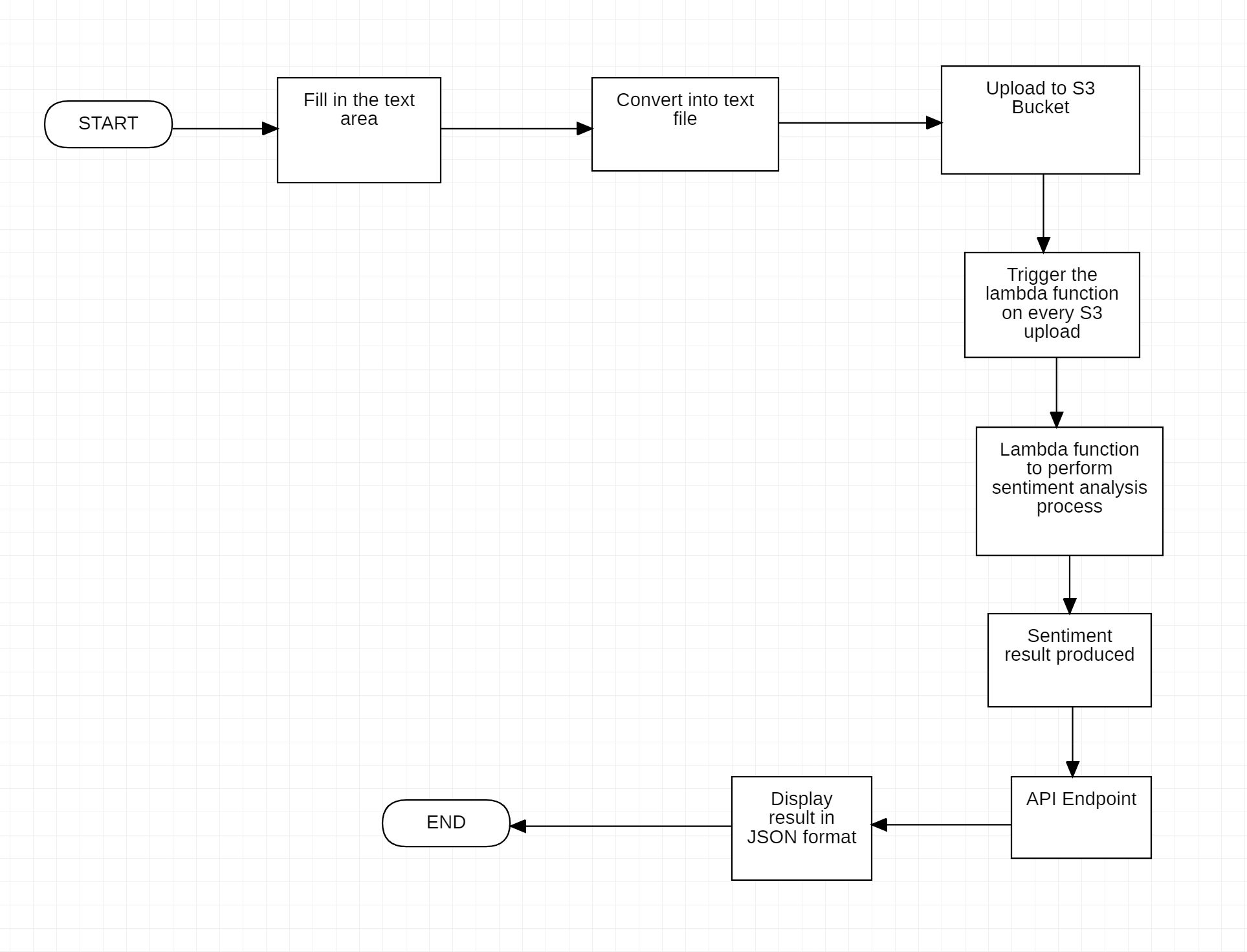
Next we come towards the backend i.e., processing part wherein comes the role of AWS Lambda function. AWS Lambda is a serverless compute service that lets you run code without provisioning or managing servers. Three libraries namely boto3, json and urllib.parse have been imported.

Boto3 library is essential for defining the s3 object and handling operations such as reading the content of the recently uploaded file. Also the comprehend functionality is invoked by using the boto3 object.

JSON is used for handling the response that is displayed in JSON format and essentially helps in extracting and displaying the useful data from the JSON response body and printing it onto the cloudwatch logs.

The urllib.parse library has utilities which are used for capturing the s3 bucket name and the file name that has been recently uploaded into the S3 bucket.

After setting up the lambda function, the next step is to integrate it with an API endpoint so that the sentiment score can be obtained and displayed outside the cloud environment. We are making a POST method API endpoint.



**Figure 2** : Flowchart for the technical implementation of Textual Emotion Recognition

* Implementation and Connection with the application

The actual mechanism of text file transfer goes as follows. The text given as input in the text area inside the application goes to the backend to the server. Here in the backend the combined application of Node.js and AWS SDK will handle the connection of the cloud environment with the application.

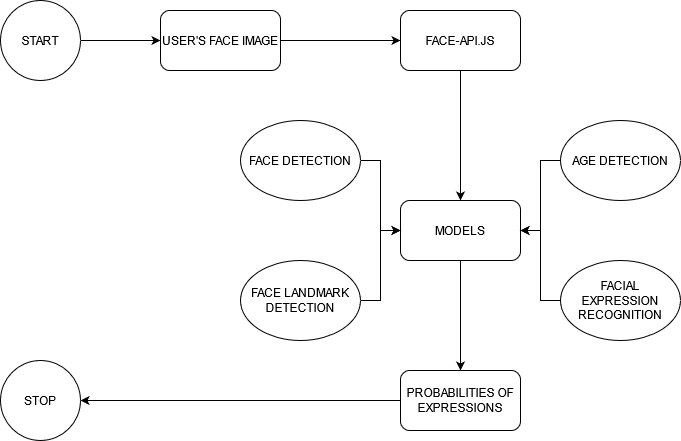
AWS SDK is required for creating the S3 bucket object and accessing the target bucket by having the access of credentials namely Access key and Secret Access Key.

The text entered in the frontend is converted into a .txt file by use of express js and the putObject function is exploited to transfer the file into the bucket. Also the random function is used for attaching a random id to the text file name and is assigned to the key parameter.

## Facial Emotion Recognition

The main key feature of this module is to detect the face of the expression of the user and generate the sentiments based on his facial expressions. All this important feature can be achieved using the face-api.js. Face-api.js is an open source javaScript Face recognition API for browser and NodeJS built on top of tensorflow.js.

Face-api.js has many different pretrained models like Face Detection, Face Landmark Detection, Face Recognition, Face Expression Recognition and Age Estimation & Gender Recognition. We have only used the Face Expression Recognition model to get the users sentiments in the real time and some computation needs to be done to generate the sentiment score.



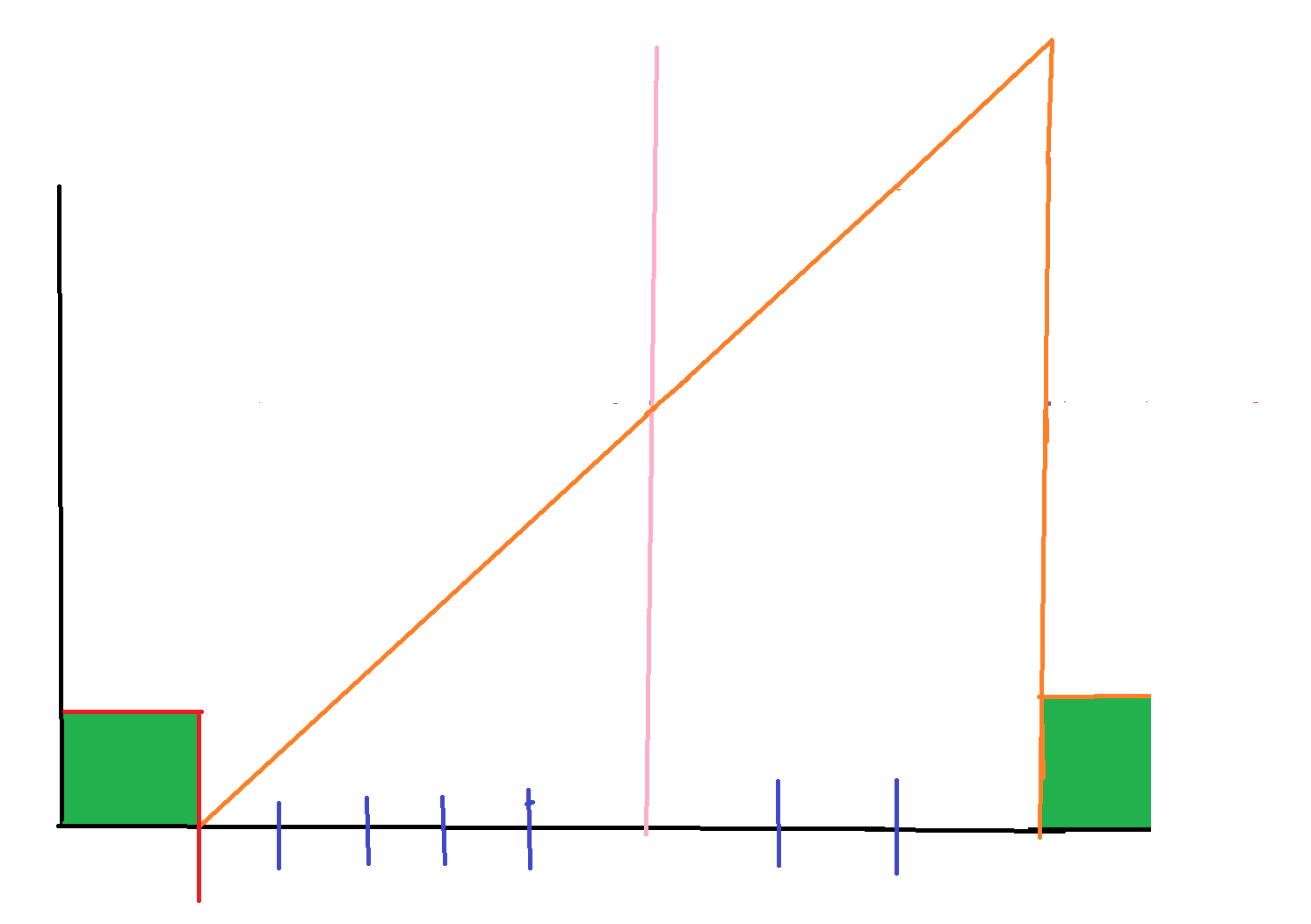
**Figure 3** : Flowchart for the technical implementation of Facial Emotion Recognition

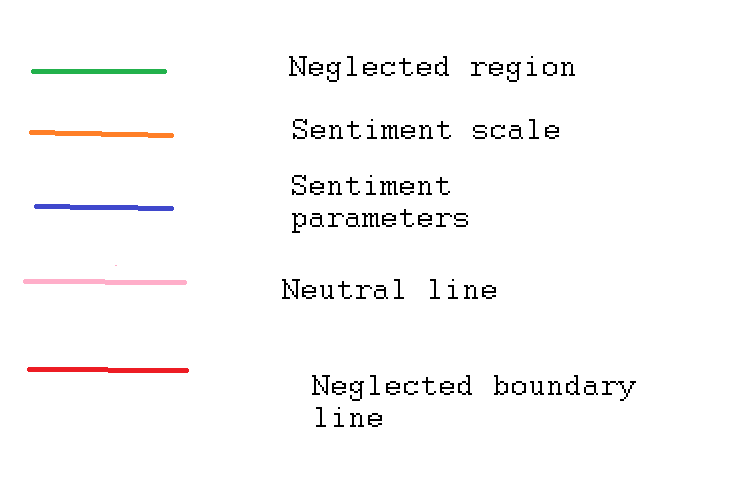
New sentiments are recognized by every new frame captured by the API. The sentiments are returned in the form of JSON with probability of certain expressions on the user's face, namely happy, surprised, neutral, sad, fearful, angry and disgusted. So at every new frame all the probability values of the sentiments are added to the corresponding variables again in JSON and when the user wants to quit the process of calibrating the sentiment based on his/her facial expression, this calculated JSON is passed to Node server and stored on the cloud.

## The Analysis

1. Sentiments score

For generalizing the standards for the facial sentiment values and textual sentiment values we have come up with a sentiment score methodology. The facial sentiment values contain seven expressions namely happy, surprised, neutral, disgusted, fearful, angry, and sad.

**Figure 4** : Representative diagram for Sentiment Score algorithm



Hence, to compute the score we have first organized the sentiments on the basis of priority positive sentiment having highest priority and negative sentiment having least priority and the priority we can up from highest to lowest is happy, surprised, neutral, disgusted, fearful, angry, sad. The values of each expression are in terms of probability hence their value ranges from 0 to 1. Now we need to add some weights to these expressions on the basis of the priority. The values of these weights would range from 0 to 1and higher weights would be given to higher priority expressions. Now to reduce the higher bias on higher priority expression and lower bias on lower priority expression the top and bottom ten percent of weight would be tailored so that the remaining range of weights would be from 0.1 to 0.9. The range would be divided into seven parts which will be yielding 8 segments. Only considering the top of these segments, the weights for the expression would be 0.22, 0.33, 0.44 and so on till 0.88 from lower to higher priority.

| Expression | Corresponding weights |
| --- | --- |
| Happy | 0.88 |
| Surprised | 0.77 |
| Neutral | 0.66 |
| Disgusted | 0.55 |
| Fearful | 0.44 |
| Angry | 0.33 |
| Sad | 0.22 |

**Table 1** : Expression v/s Corresponding weights

Similarly, the textual sentiment values contain four expressions namely positive, negative, neutral and mixed. The number of parameters in the facial sentiment values is seven but in the textual sentiment values is four so we need to scale up these four parameters to the seven parameters of the facial sentiment values.

(i) Scaling up the positive sentiment value:

Out of all the expression parameters obtained as a result of facial sentiment analysis, happy and surprised are the only positive expressions and therefore can be broadly categorized into positive expression parameters of the textual sentiment values.

here 0.88 represents the weight of happy sentiment and 0.77 represents the weight of surprised sentiment. The happy sentiment value is obtained by multiplying the positive sentiment value with the ratio obtained by the weight of happy expression to the summation of weights of happy and surprised expressions.

here 0.88 represents the weight of happy sentiment and 0.77 represents the weight of surprised sentiment. The surprised sentiment value is obtained by multiplying the positive sentiment value with the ratio obtained by the weight of surprised expression to the summation of weights of happy and surprised expressions.

(ii) Scaling up the neutral sentiment value:

The neutral sentiment value obtained as a result of facial sentiment analysis can directly be mapped to the neutral sentiment value obtained as a result of the textual sentiment analysis.

here the neutral expression value of the textual sentiment expression is the same as the neutral expression value of the facial sentiment expression.

(iii) Scaling up the negative sentiment value:

Out of all the expression parameters obtained as a result of facial sentiment analysis, disgusted, fearful, angry and sad are the only negative expressions and therefore can be broadly categorized into negative expression parameters of the textual sentiment values.

Here 0.55 represents the weight of disgusted expression and 0.55, 0.44, 0.33, 0.22 represents the corresponding weights of disgusted, fearful, angry and sad respectively. The weight of disgusted sentiment value is obtained by multiplying the negative sentiment value with the ratio obtained by weight of disgusted expression to the summation of weights of disgusted, fearful, angry and sad expressions.

Here 0.55 represents the weight of fearful expression and 0.55, 0.44, 0.33, 0.22 represents the corresponding weights of disgusted, fearful, angry and sad respectively. The weight of disgusted sentiment value is obtained by multiplying the negative sentiment value with the ratio obtained by the weight of fearful expression to the summation of weights of disgusted, fearful, angry and sad expressions.

Here 0.33 represents the weight of angry expression and 0.55, 0.44, 0.33, 0.22 represents the corresponding weights of disgusted, fearful, angry and sad respectively. The weight of angry sentiment value is obtained by multiplying the negative sentiment value with the ratio obtained by the weight of angry expression to the summation of weights of disgusted, fearful, angry and sad expressions.

Here 0.22 represents the weight of sad expression and 0.55, 0.44, 0.33, 0.22 represents the corresponding weights of disgusted, fearful, angry and sad respectively. The weight of sad sentiment value is obtained by multiplying the negative sentiment value with the ratio obtained by the weight of sad expression to the summation of weights of disgusted, fearful, angry and sad expressions.

(iv) Scaling up the mixed sentiment value

The mixed sentiment value is being ignored as it is ambiguous and does not fit the above proposed algorithm.

2. Visualization for the Facial Emotional Recognition

Facial expression parameters like happy, surprised, neutral, angry, disgusted, fearful, and sad are captured as long as the webcam stream continues. This data is then stored in another Javascript object. Along with this, the count of frames is also captured for facilitating average calculation. Parameter with the maximum magnitude becomes the resultant emotion. The data object thus created is then passed into the session storage so that the data then extracted can be used for visualization purposes.

The final result of Textual Emotional Recognition is available in the API endpoint in JSON format. This data is then extracted by making use of JQuery and the values corresponding to the parameters namely positive, negative, mixed and neutral are passed into the session storage after being stored in a javascript object.

Finally the plotting of the dynamic pie chart is done using google static chart API for neat, responsive and organised pie charts.

## Module Integration and Interconnection

#### **Authorize and Authentication**

Authorizing and authenticating the user is very crucial and important in maintaining the user’s data readily available and anonymous to other users on the application. There are many dependencies readily available for NodeJS server. One used in this approach is Passport.js which stores the current users data in the memory and not in the local storage or session storage which removes the possibility of any XSS attacks.The user will only be able to use the application if they register with a unique email and login in using the same registered email and password. All the users will be identified by a unique user ID and all the data related to that user will only be identified using this user ID.

1. **Database**

The data generated by the users could be tremendous hence NoSQL database is preferred over traditional SQL database. NoSQL can greatly reduce the data extraction time. In this case, MongoDB is used as a NoSQL cloud database to store all the users data. Mongoose is used for connecting the NodeJS server to MongoDB. Mongoose provides a straight-forward, schema-based solution to model the application data. It includes built-in type casting, validation, query building and etc. And Joi was used to validate the schemas before sending to the database.

1. **Textual analysis**

Concerning the frontend, the input text is entered inside a text area given in the application. This text is then bundled and sent to the backend server which is managed by NodeJs. This is the section wherein the integration of cloud environment with the real time application is established. Finally with the help of AWS bucket credentials, the target bucket is accessed. Here the input text is converted into a .txt file with some uniquely identified file name so as to send it to the S3 bucket.

The processing is done in the cloud environment and now the output is transferred from the cloud environment to the backend by means of a POST API endpoint constructed using the API Gateway service.This POST request will trigger the AWS lambda function to process and analyze the sentiments of the given text file and the response body will contain the sentiments of that file. This response is then handled at the NodeJS server.

1. **Facial Analysis**

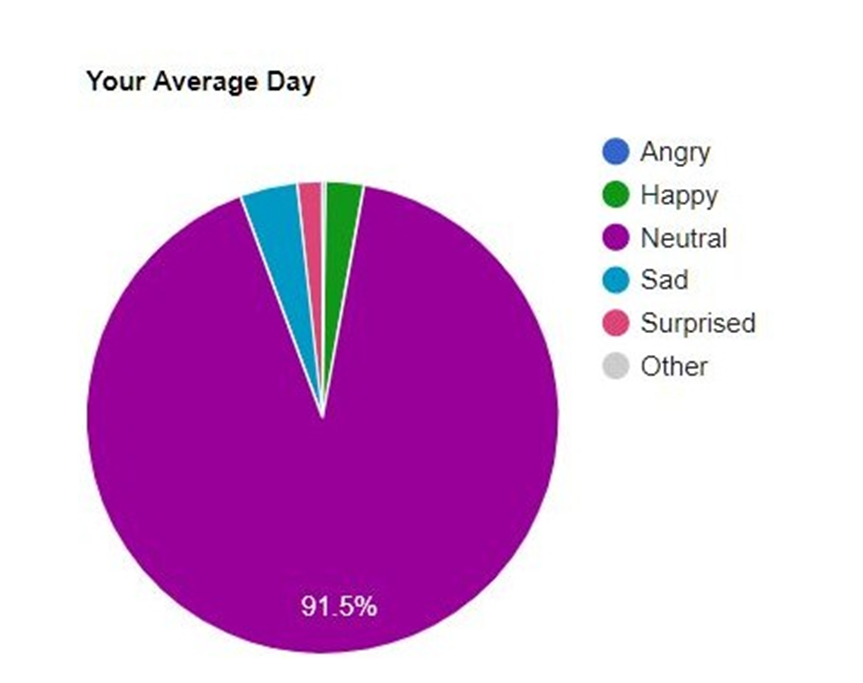
When the user enters the facial analysis section, a permission to access the webcam would pop up. After allowing the access, a webcam stream would start which would capture images at a regular interval of time (2-3 seconds). Simultaneously face-api.js will start image processing and calculate the probability of each sentiment along with recording the probability of the sentiment. The count of each frame would also be recorded and this recorded data is then sent to the NodeJS server via POST request. Furthermore the unique user ID is added to the above data which in turn is stored in the MongoDB database.

1. **Analysis**

NodeJS will fetch all the textual and facial sentiment values of a particular user from the database and this data will be sent to the frontend of that particular user.

The data thus obtained on the frontend will be used for two purposes namely data visualization and sentiment score calculation. But prior to this data preprocessing is needed because textual sentiment has four parameters whereas the facial sentiment has seven parameters. The algorithm used for achieving data pre-processing has been discussed as postulate number 1 of section F under the Ease of Use.

Now the visualization of this preprocessed data is achieved by giving it as input to the Google Chart API.



**Figure 5** : Pie chart visualisation

Now the other part of the analysis is computing the sentiment score based on the weights of the sentiments (refer table 1). The formula to compute the sentiment score is

Sentiment score = (expression weight\*expression value)

Sentiment score represents the summation of the product of the expression weights and expression values corresponding to each sentiment parameter.

* If the sentiment score is less than 0.605 then the expression is considered to be negative.
* If the sentiment score is greater than 0.605 and less than 0.715 then the expression is considered as neutral
* If the sentiment score is greater than 0.715 then the expression is considered as positive

## Future Scope

As a future scope predictive models might turn out to be an efficient solution to get insight into the data thus collected after the sentiment detection through both the modes. Given the format of current data we are dealing with sentiment scores corresponding to various emotions which can then be broadly sub categorized into positive, neutral and negative.

Also as an additional feature we aim to make a prediction about the sentiment an individual might experience in the near future based on calculations drawn from the data predictive models.

##### CONCLUSION

In this way two prominent parameters namely facial and textual expression sentiment values have been successfully utilised to get meaningful observations which could help in overcoming the problems like depression and other psychological ailments. The insights into the data thus gathered also helped in generating innovative analysis techniques.

The finished solution can be viewed at : <https://github.com/kunatastic/AWS-AI-ML/>

##### Acknowledgment

We thank our Internet and Web Programming faculty Dr. Sandhya P from Vellore Institute of Technology , Chennai who provided expertise and insight that proved to be very helpful.

##### References

(i) Setting up S3 trigger for lambda function :

<https://docs.aws.amazon.com/lambda/latest/dg/with-s3-example.html>

(ii) Use of AWS Comprehend in lambda function :

<https://aws.amazon.com/blogs/compute/using-aws-lambda-and-amazon-comprehend-for-sentiment-analysis/>

(iii)Making the API Gateway :

<https://www.freecodecamp.org/news/building-an-api-with-lambdas-and-api-gateway-11254e23b703/>

(iv)Passport.js documentation:

<http://www.passportjs.org/docs/>

(v)Mongoose documentation

<https://mongoosejs.com/docs/guide.html>

(vi)Joi documentation

<https://joi.dev/api/?v=17.4.0>

(vii) Face-api.js

<https://github.com/justadudewhohacks/face-api.js>