Final Report
on

Event Analysis

Submitted for partial fulfillment of award of

BACHELOR OF TECHNOLOGY

degree

In

Computer Engineering & Information Technology

By

Anushruti (1803211014)
Richa Singh (1803211036)
Shrey Kr. Saxena (1803211044)

Mrs. Abhilasha
SUPERVISOR

ABES ENGINEERING COLLEGE, GHAZIABAD,

Dr. A P J ABDUL KALAM TECHNICAL UNIVERSITY
LUCKNOW
June 2022

Certificate

This is certified that **Anushruti, Shrey kr. Saxena, Richa Singh** has carried out the project work presented in this report entitled **“Event Analysis usimg deep learning”** for the partial fulfillment of **Bachelor of Technology** from, ABEC Engineering college under my supervision. The report embodies result of original work and studies carried out by Student himself/herself and the contents of the report do not form the basis for the award of any other degree to the candidate or to anybody else.

Signature:

Designation:

Address:

Date:

Acknowledgement

We take this opportunity to thank our teachers and friends who helped us throughout the project. First and foremost, I would like to thank my guide (Mrs. Abhilasha and Ms. Tanya Gupta) for her valuable advice and time during development of project. We would also like to thank **Dr. Pankaj Kumar Sharma (HOD, Computer Science Department**) for his constant support during the development of the project.

Name: Anushruti
Roll No. : 1803211014
Signature:

Name: Richa Singh
RollNo:1803211036
Signature:

Name: Shrey Kr. Saxeena
Roll No. : 1803211044
Signature:

Declaration

I hereby declare that this submission is my own work and that, to the best of my knowledge and belief, it contains no material previously published or written by another person nor material which to a substantial extent has been accepted for the award of any other degree or diploma of the university or other institute of higher learning, except where due acknowledgment has been made in the text.

Signature:
Name:
Roll No.
Date:

Signature:
Name:
Roll No.
Date:

Signature:
Name:
Roll no
Date:

Abstract

In the current scenario we have seen that there has been ,many developments of various deep learning techniques, that have been proven successful in various domains I.e. audio, image and NLP (natural language processing).deep learning helps in understanding the structure and Semantic of sentences. This trend is growing nowadays and many researchers are moving away From traditional machine learning to deep learning for their scientific research. Among them face detection, speech recognition gesture recognition, biometric recognition have been studied and a lot of work has been introduced. In this paper we will display an overview on event analysis based on deep learning techniques. We will be defining emotion, emotion classification methods we summarize deep learning technology

TABLE OF CONTENT

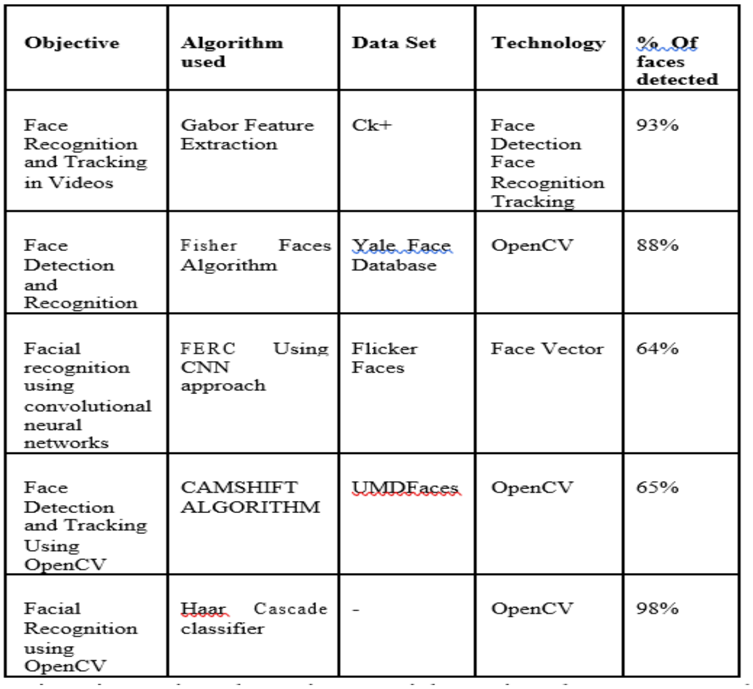
CHAPTER NO. PAGE NO.

1. INTRODUCTION

Emotion Analysis is the task of extracting and analyzing user emotional states in texts. EA Not only acts as a standalone tool for information extraction but also plays an important role for various Natural Language Processing (NLP) applications, including eCommerce [1], public opinion analysis [2], big search [3], Information prediction [4], personalized recommendation [5], healthcare [6], and online teaching [7]. We Collected papers from various sources such as ACL Web Anthology, AAAI, IEEE, ACM, Elsevier, and Springer. Face recognition is a long-standing topic in the research Community of computer vision. A standard pipeline of the Recognition framework consists of four individual steps: locating faces with bounding boxes and fiducial points, aligning face images using a pre-defined template, extracting face Representations and representation comparing. The second Step, also named as face alignment, serves as deforming face images such that fiducial points are spatially aligned and simplifies the recognition task by normalizing. The in-plane rotation, scale and translation variations. Face features can be divided into two sets depending on the zone where they are located: internal features including eyes, nose and Mouth, and external features, composed by the hair, chin and Face outline.

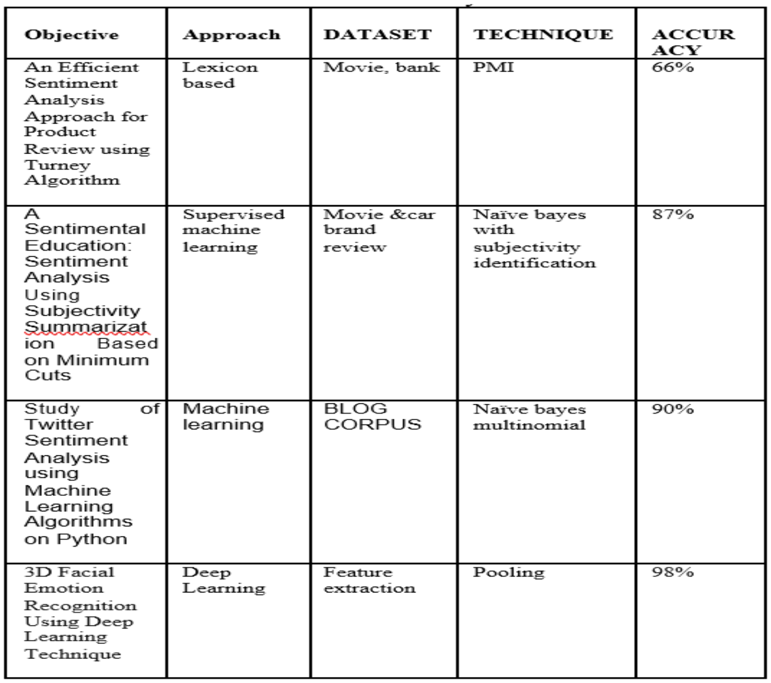
There are 6 emotion categories that are widely used to describe humans' basic emotions, based on facial expression [1]: anger, disgust, fear, happiness, sadness and surprise. These are mainly associated with negative sentiment, with "Surprise" being the most ambiguous, as it can be associated with either positive or negative feelings. Interestingly, the number of basic human emotions has been recently "reduced", or rather re-categorized, to just 4; happiness, sadness, fear/surprise, and anger/disgust.

2. LITERATURE REVIEW



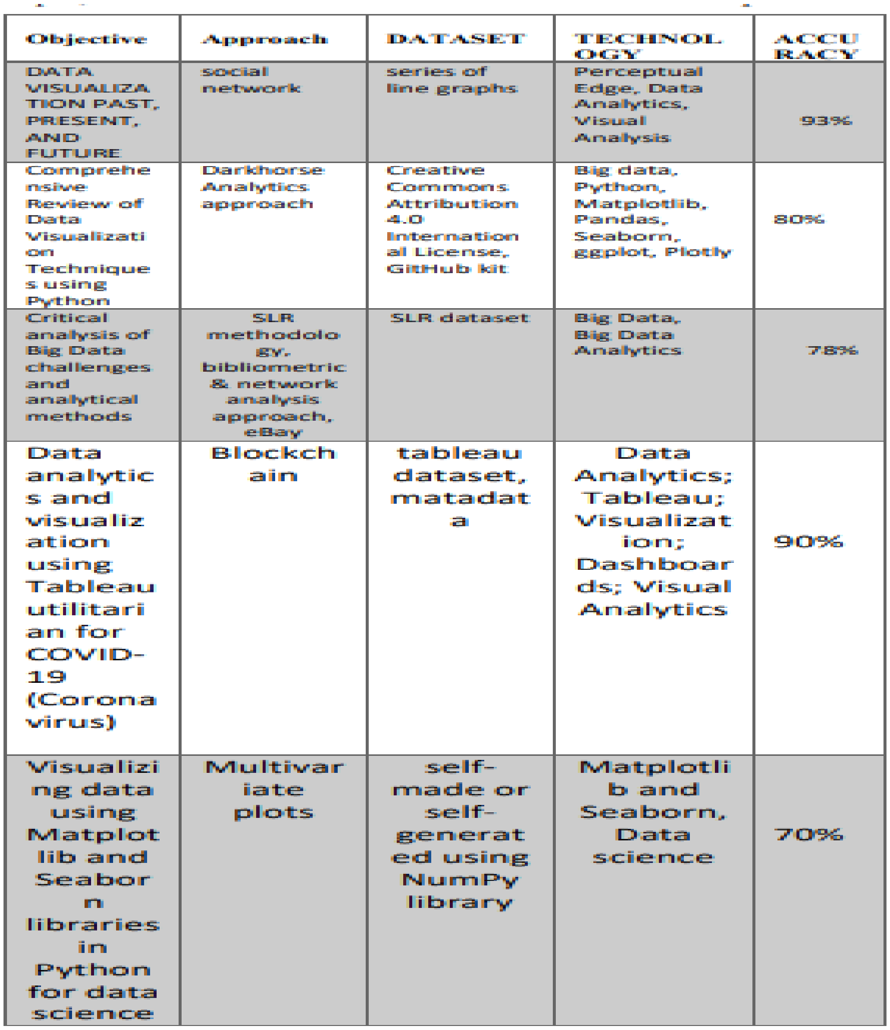
Objective	Algorithm used	Data Set	Technology	% Of faces detected
Face Recognition and Tracking in Videos	Gabor Feature Extraction	Ck+	Face Detection Face Recognition Tracking	93%
Face Detection and Recognition	Fisher Faces Algorithm	Yale Face Database	OpenCV	88%
Facial recognition using convolutional neural networks	FERC Using CNN approach	Flicker Faces	Face Vector	64%
Face Detection and Tracking Using OpenCV	CAMSHIFT ALGORITHM	UMDFaces	OpenCV	65%
Facial Recognition using OpenCV	Haar Cascade classifier	-	OpenCV	98%

Table.1. Literature review on Face Recognition



Objective	Approach	DATASET	TECHNIQUE	ACCURACY
An Efficient Sentiment Analysis Approach for Product Review using Turney Algorithm	Lexicon based	Movie, bank	PMI	66%
A Sentimental Education: Sentiment Analysis Using Subjectivity Summarization Based on Minimum Cuts	Supervised machine learning	Movie &car brand review	Naïve bayes with subjectivity identification	87%
Study of Twitter Sentiment Analysis using Machine Learning Algorithms on Python	Machine learning	BLOG CORPUS	Naïve bayes multinomial	90%
3D Facial Emotion Recognition Using Deep Learning Technique	Deep Learning	Feature extraction	Pooling	98%

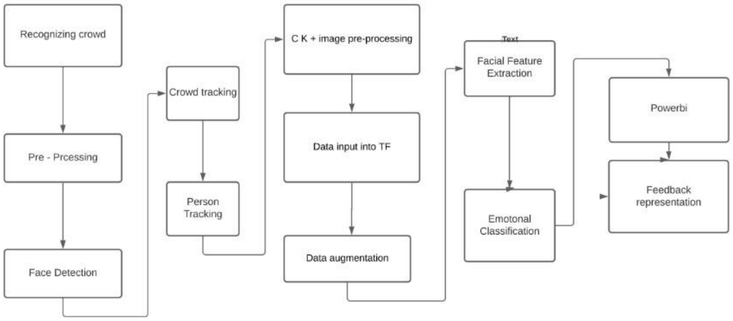
Table.2. Literature review of Emotion detection



Objective	Approach	DATASET	TECHNOLOGY	ACCURACY
DATA VISUALIZATION PAST, PRESENT, AND FUTURE	social network	series of line graphs	Perceptual Edge, Data Analytics, Visual Analysis	93%
Comprehensive Review of Data Visualization Techniques using Python	Darkhorse Analytics approach	Creative Commons Attribution 4.0 International License, GitHub kit	Big data, Python, Matplotlib, Pandas, Seaborn, ggplot, Plotly	80%
Critical analysis of Big Data challenges and analytical methods	SLR methodology, bibliometric & network analysis approach, eBay	SLR dataset	Big Data, Big Data Analytics	78%
Data analytics and visualization using Tableau utilitarian for COVID-19 (Coronavirus)	Blockchain	tableau dataset, matadata	Data Analytics; Tableau; Visualization; Dashboards; Visual Analytics	90%
Visualizing data using Matplotlib and Seaborn libraries in Python for data science	Multivariate plots	self-made or self-generated using NumPy library	Matplotlib and Seaborn, Data science	70%

Table.3. Literature review of Data Analysis

3. METHODOLOGY



A. Recognizing faces in Crowd

The first step in our pipeline is face detection. Obviously we need to locate the faces in a crowd before we can try to tell them apart! Face Detection A proper and efficient face detection algorithm always increases the performance of face recognition systems. Various algorithms are proposed for face detection such as face knowledge-based methods, feature invariant methods, machine learning based methods. In this project, Firstly we implemented a system for locating faces in crowd. Before we continue, we must differentiate between face recognition and face detection. They are not the same, but one depends on the other. In this case face recognition needs face detection for making an identification to "recognize" a face. We will only cover face detection. Face detection uses classifiers, which are algorithms that detects what is either a face(1) or not a face(0) in an image. Classifiers have been trained to detect faces using thousands to millions of people in order to get more accuracy. OpenCV uses two types of classifiers, LBP (Local Binary Pattern) and Haar Cascades Understanding Haar Cascades A Haar Cascade is based on "Haar Wavelets" which defines as: A sequence of rescaled "square-shaped" functions which together form a wavelet family or basis. It is based on the Haar Wavelet technique to analyze pixels in the image into squares by function. This uses machine learning techniques to get a high degree of accuracy from what is called "training data". This uses "integral image" concepts to compute the "features" detected. Haar Cascades use the Adaboost learning algorithm which selects a small number of important features from a large set to give an efficient result of classifiers.

B. Feature Extraction (Pre-processing)

Haar Cascades use machine learning techniques in which a function is trained from a lot of positive and negative images. This process in the algorithm is feature extraction. In this Biometric based system collection of every individual is required. This database development phase consists of image capture of every audience, in our proposed system it is face, and after it is enhanced using pre-processing techniques and to be stored in the database.

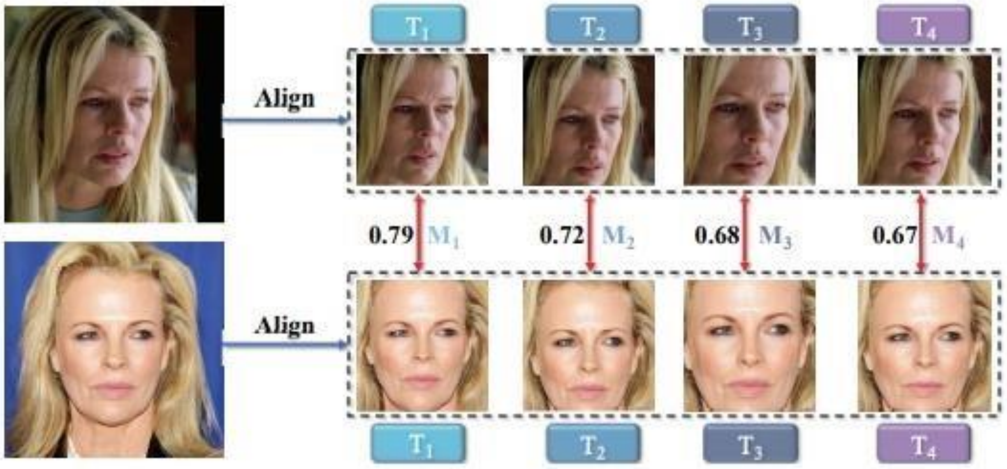


Fig.1. It explains the steps of feature extraction one by one

C. Data input into database (TF)

Data input into TF On this section, the first interaction with TF is encountered. The binary file contraining the images and their corresponding label is feed into a record reader. This operation created a queue consisted of all the training examples.

- filenames = [os.path.join(PATH, FILENAME)]
- Create a queue that produces the filenames to read.
- filename queue = tf.tr a in.string in put producer(filenames)
- # Read examples from files in the filename queue.
- read input = read(filename queue) .

After that some cast operation are performed on the key and value.

Key: The value of the label after being cast as an int32. ·

Value: The tensor containing the tensor representing the image.

Later, the tensor is reshaped from depth, height and width to height, width, and depth. Finally, it is time to apply transformations into the image to perform data augmentation.

D. Data Augmentation

The bigger the dataset, the more complex to label it. Deep Learning requires big amounts of data for training. Since this is a very expensive task, data augmentation has been proven an efficient way to expand the dataset Data augmentation consists of applying transformation on the corpus. In this case, transformations were applied over CK+ images. Modifying image properties helps to exploit the invariant features that the network can learn. TF provides a set of functions suitable for image transformation. The topology to be used is inspired by the one developed on the Visual Geometry Group (VGG) at University of Oxford. VGG network was awarded as the runner-up on ImageNet 2014's competition, and it stressed network depth's importance to improve classification accuracy. After finishing the image processing, the next step is to create image batches to feed the network. The batch size is another parameter to take into account. Since the network optimization is performed by stochastic gradient descend (SGD) with momentum, choosing a correct batch size is critical. Some aspects to take into account to set the batch size are the data set size, and hardware availability. It would be optimal to take into account the whole data set in each step to optimize towards the gradient; however such an approach would be computationally expensive and time consuming [102]. It is common to find on the literature that the batch size is a power of two. For instance, on Krizhevsky et al. [1], the batch size is 128. In this project, the batch size was set to 64. This value was defined after trying several values. It provided a good trade-off between the training time and loss reduction

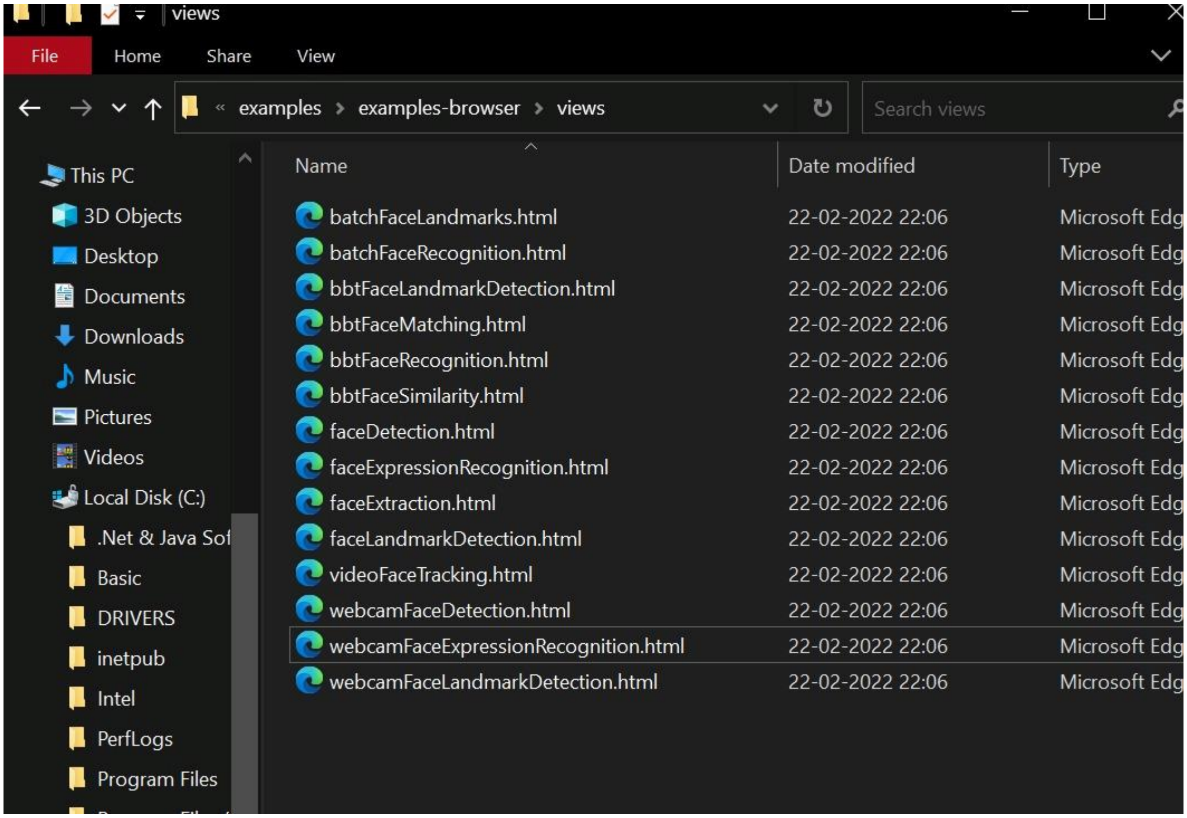
E. Training

In this section, the training configuration is displayed: the cost function and the learning rate. Cross entropy The training objective is to adjust weights and biases values on the network's layers. This is done by minimizing a cost function. The cost function is the cross-entropy be-tween the original labels and the model's prediction. This function is chosen because the last layer returns a distribution over all the 6 labels. Cross-entropy is a specifically error measurement for probability distributions. The fact that each image has one and only one label (mutually exclusive labels) was exploited to use cross entropy The next step is to update weights and biases values accordingly. While the cross entropy loss is minimized, it moves the weights towards the gradients.

F. Analyzing the sentiments and representing data

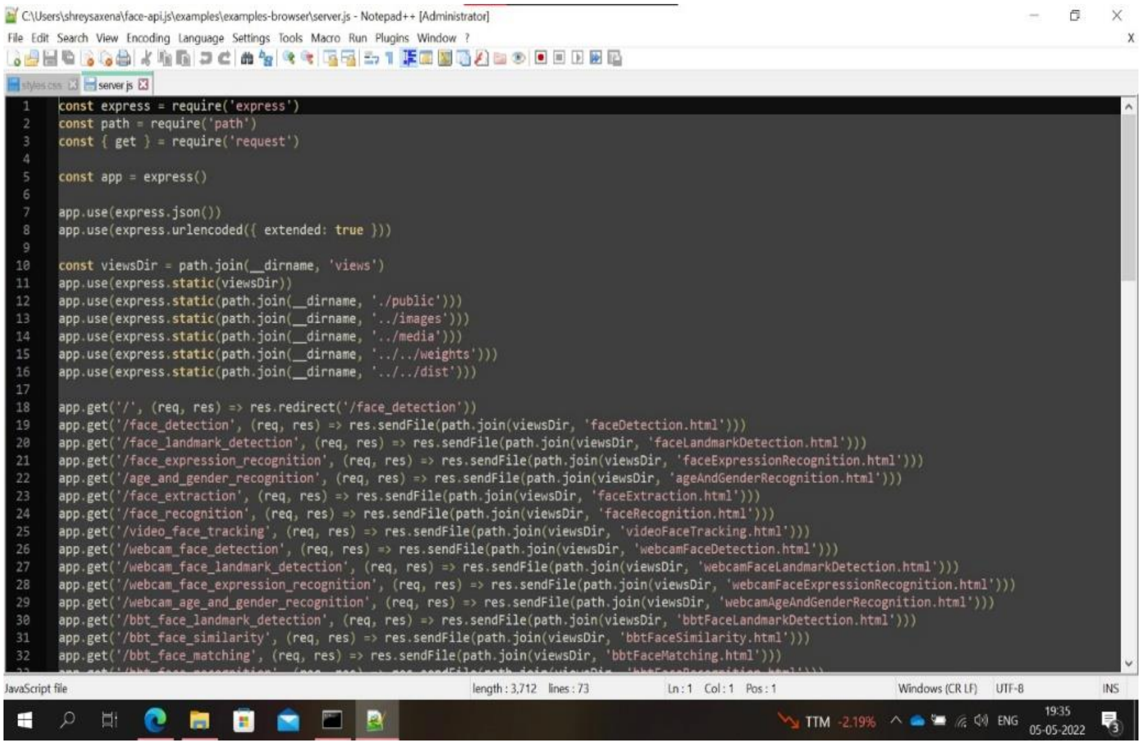
For representation of data, we use Power BI. They visualize the data in more catchy way. The analysis of the emotions of an audience will be later on visualized using Power BI.

4. IMPLEMENTATION



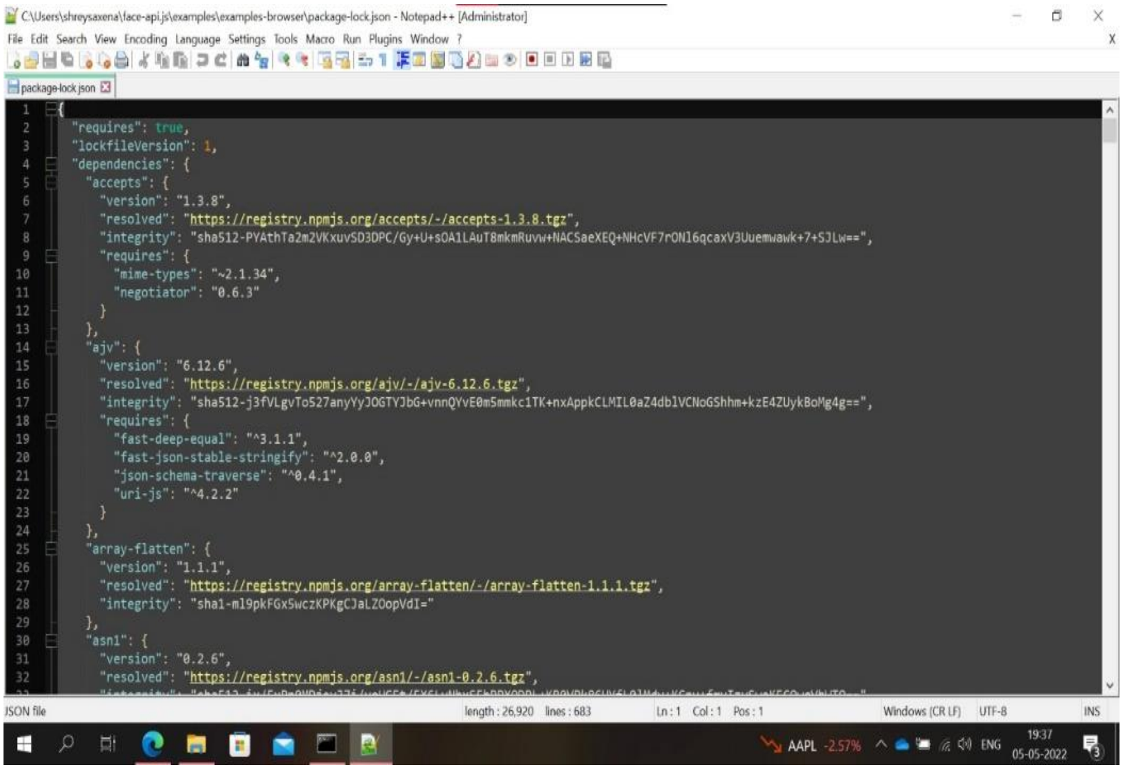
Name	Date modified	Type
batchFaceLandmarks.html	22-02-2022 22:06	Microsoft Edg
batchFaceRecognition.html	22-02-2022 22:06	Microsoft Edg
bbtFaceLandmarkDetection.html	22-02-2022 22:06	Microsoft Edg
bbtFaceMatching.html	22-02-2022 22:06	Microsoft Edg
bbtFaceRecognition.html	22-02-2022 22:06	Microsoft Edg
bbtFaceSimilarity.html	22-02-2022 22:06	Microsoft Edg
faceDetection.html	22-02-2022 22:06	Microsoft Edg
faceExpressionRecognition.html	22-02-2022 22:06	Microsoft Edg
faceExtraction.html	22-02-2022 22:06	Microsoft Edg
faceLandmarkDetection.html	22-02-2022 22:06	Microsoft Edg
videoFaceTracking.html	22-02-2022 22:06	Microsoft Edg
webcamFaceDetection.html	22-02-2022 22:06	Microsoft Edg
webcamFaceExpressionRecognition.html	22-02-2022 22:06	Microsoft Edg
webcamFaceLandmarkDetection.html	22-02-2022 22:06	Microsoft Edg

Fig.3. server.js file


C:\Users\shreysaxena\face-api.js\examples\examples-browser\server.js - Notepad++ [Administrator]
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
styles.css
server.js
JavaScript file
length : 3,712 lines : 73
Ln : 1 Col : 1 Pos : 1
Windows (CR LF) UTF-8 INS
TTM -2.19%
ENG
19:35
05-05-2022

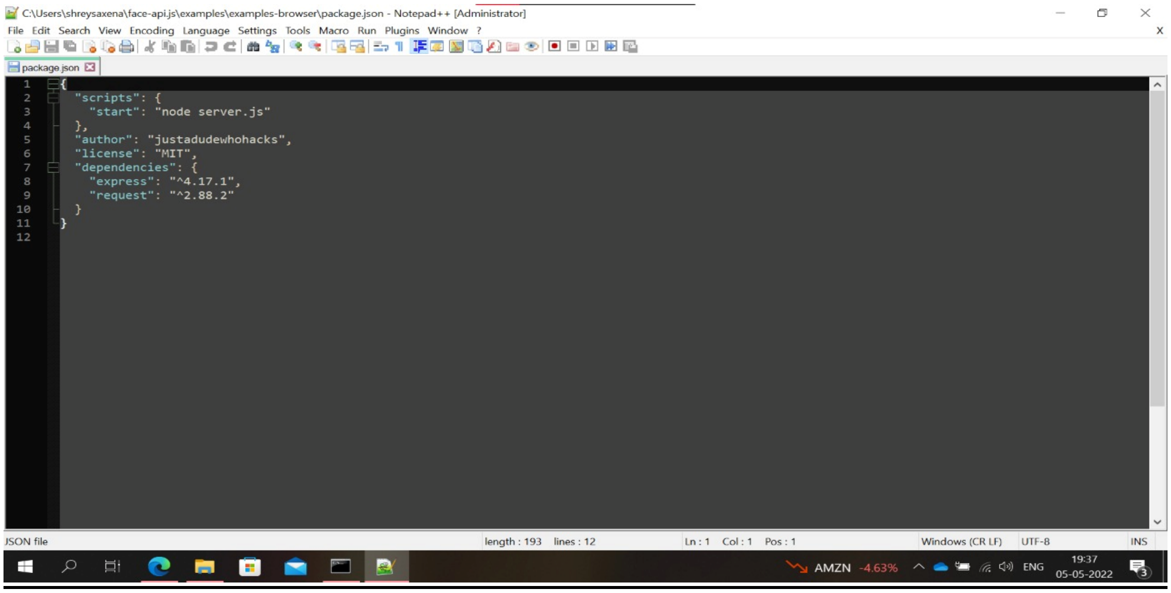
```
const express = require('express')
const path = require('path')
const { get } = require('request')

const app = express()

app.use(express.json())
app.use(express.urlencoded({ extended: true }))

const viewsDir = path.join(__dirname, 'views')
app.use(express.static(viewsDir))
app.use(express.static(path.join(__dirname, './public')))
app.use(express.static(path.join(__dirname, '../images')))
app.use(express.static(path.join(__dirname, '../media')))
app.use(express.static(path.join(__dirname, '../../weights')))
app.use(express.static(path.join(__dirname, '../../dist')))

app.get('/', (req, res) => res.redirect('/face_detection'))
app.get('/face_detection', (req, res) => res.sendFile(path.join(viewsDir, 'faceDetection.html')))
app.get('/face_landmark_detection', (req, res) => res.sendFile(path.join(viewsDir, 'faceLandmarkDetection.html')))
app.get('/face_expression_recognition', (req, res) => res.sendFile(path.join(viewsDir, 'faceExpressionRecognition.html')))
app.get('/age_and_gender_recognition', (req, res) => res.sendFile(path.join(viewsDir, 'ageAndGenderRecognition.html')))
app.get('/face_extraction', (req, res) => res.sendFile(path.join(viewsDir, 'faceExtraction.html')))
app.get('/face_recognition', (req, res) => res.sendFile(path.join(viewsDir, 'faceRecognition.html')))
app.get('/video_face_tracking', (req, res) => res.sendFile(path.join(viewsDir, 'videoFaceTracking.html')))
app.get('/webcam_face_detection', (req, res) => res.sendFile(path.join(viewsDir, 'webcamFaceDetection.html')))
app.get('/webcam_face_landmark_detection', (req, res) => res.sendFile(path.join(viewsDir, 'webcamFaceLandmarkDetection.html')))
app.get('/webcam_face_expression_recognition', (req, res) => res.sendFile(path.join(viewsDir, 'webcamFaceExpressionRecognition.html')))
app.get('/webcam_age_and_gender_recognition', (req, res) => res.sendFile(path.join(viewsDir, 'webcamAgeAndGenderRecognition.html')))
app.get('/bbt_face_landmark_detection', (req, res) => res.sendFile(path.join(viewsDir, 'bbtFaceLandmarkDetection.html')))
app.get('/bbt_face_similarity', (req, res) => res.sendFile(path.join(viewsDir, 'bbtFaceSimilarity.html')))
app.get('/bbt_face_matching', (req, res) => res.sendFile(path.join(viewsDir, 'bbtFaceMatching.html')))
```

Fig.4 server.js file


C:\Users\shreysaxena\face-api.js\examples\examples-browser\package-lock.json - Notepad++ [Administrator]
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
package-lock.json
JSON file
length : 26,920 lines : 683
Ln : 1 Col : 1 Pos : 1
Windows (CR LF) UTF-8 INS
AAPL -2.57%
ENG
19:37
05-05-2022

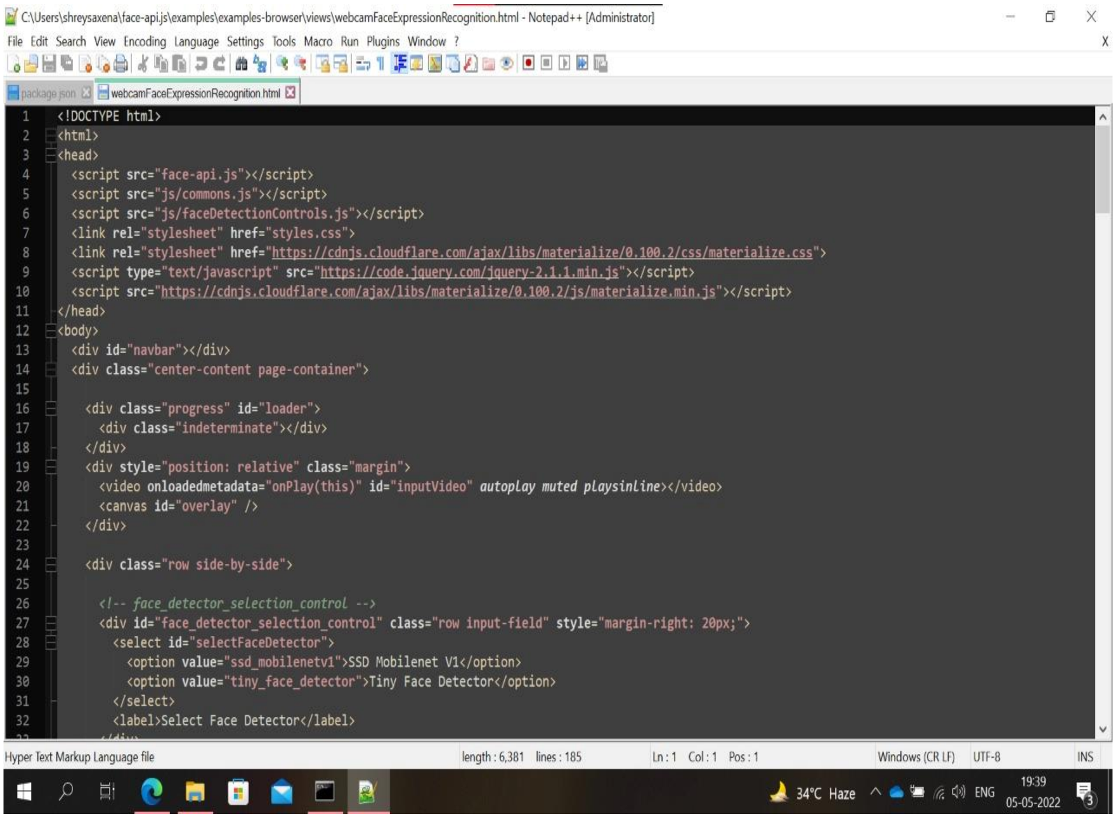
```
{
  "requires": true,
  "lockfileVersion": 1,
  "dependencies": {
    "accepts": {
      "version": "1.3.8",
      "resolved": "https://registry.npmjs.org/accepts/-/accepts-1.3.8.tgz",
      "integrity": "sha512-PYAthTa2m2VKxuvSD3DPC/Gy+U+sOA1LAuT8mkmRuvw+NACSaeXEQ+NHcVF7rONl6qcaxV3Uuemwawk+7+SJLw==",
      "requires": {
        "mime-types": "~2.1.34",
        "negotiator": "0.6.3"
      }
    },
    "ajv": {
      "version": "6.12.6",
      "resolved": "https://registry.npmjs.org/ajv/-/ajv-6.12.6.tgz",
      "integrity": "sha512-j3fVLgvTo527anyYyJOGTYJbG+vnnQYvE0m5mmkc1TK+nxAppkCLMIL0aZ4dblVCNoGShhm+kzE4ZUykBoMg4g==",
      "requires": {
        "fast-deep-equal": "^3.1.1",
        "fast-json-stable-stringify": "^2.0.0",
        "json-schema-traverse": "^0.4.1",
        "uri-js": "^4.2.2"
      }
    },
    "array-flatten": {
      "version": "1.1.1",
      "resolved": "https://registry.npmjs.org/array-flatten/-/array-flatten-1.1.1.tgz",
      "integrity": "sha1-ml9pkFGx5wczKPKgCJaLZOopVdI="
    },
    "asn1": {
      "version": "0.2.6",
      "resolved": "https://registry.npmjs.org/asn1/-/asn1-0.2.6.tgz",
```

Fig.5. packadgelock.json(i)


C:\Users\shreysaxena\face-api.js\examples\examples-browser\package.json - Notepad++ [Administrator]
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
package.json
JSON file
length : 193 lines : 12
Ln : 1 Col : 1 Pos : 1
Windows (CR LF) UTF-8 INS
AMZN -4.63%
ENG
19:37
05-05-2022

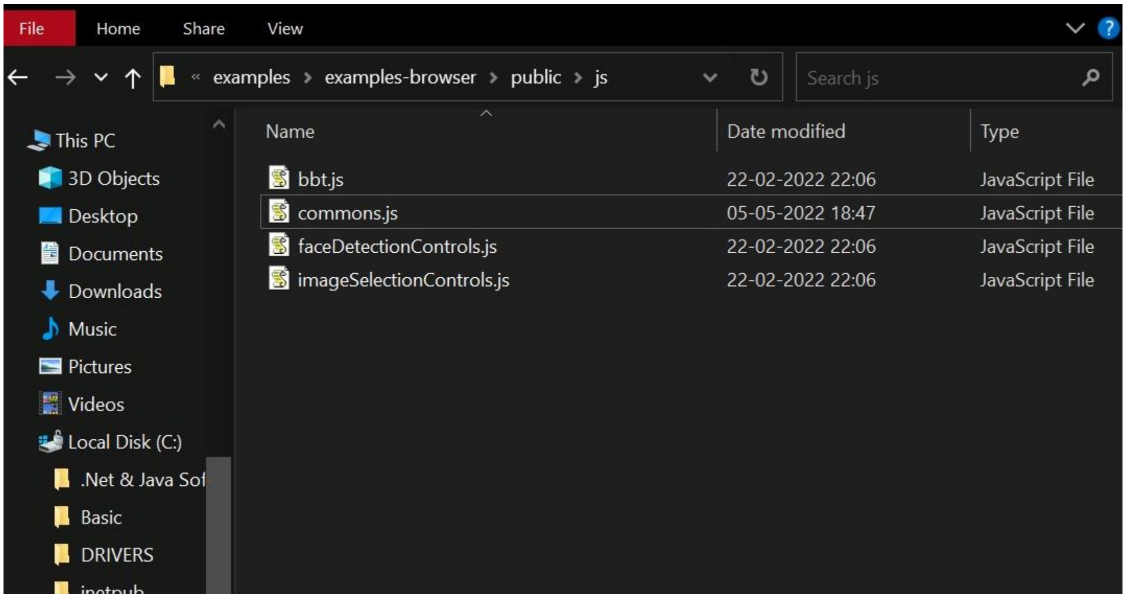
```
{
  "scripts": {
    "start": "node server.js"
  },
  "author": "justadudewhohacks",
  "license": "MIT",
  "dependencies": {
    "express": "^4.17.1",
    "request": "^2.88.2"
  }
}

```

Fig.6. packadgelock.json(ii)


C:\Users\shreysaxena\face-api.js\examples\examples-browser\views\webcamFaceExpressionRecognition.html - Notepad++ [Administrator]
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
package.json
webcamFaceExpressionRecognition.html
Hyper Text Markup Language file
length : 6,381 lines : 185
Ln : 1 Col : 1 Pos : 1
Windows (CR LF) UTF-8 INS
34°C Haze
ENG
19:39
05-05-2022

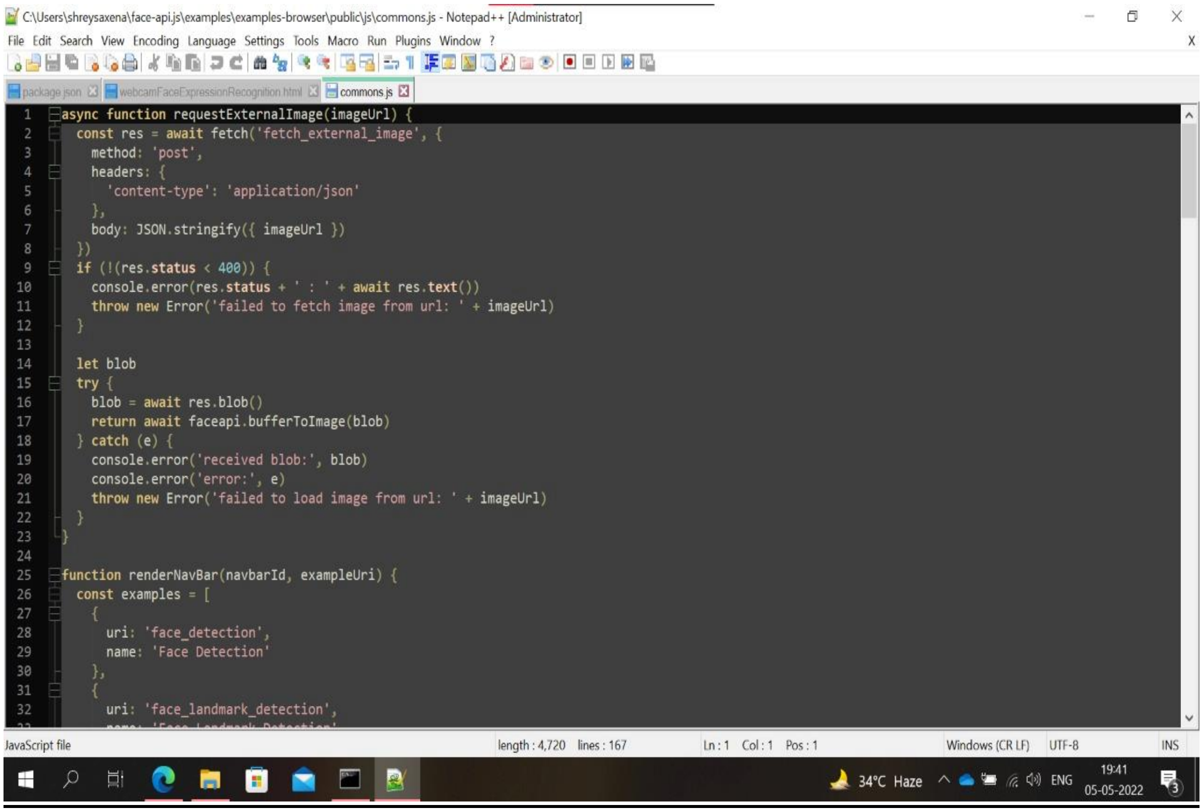
```
<!DOCTYPE html>
<html>
<head>
  <script src="face-api.js"></script>
  <script src="js/commons.js"></script>
  <script src="js/faceDetectionControls.js"></script>
  <link rel="stylesheet" href="styles.css">
  <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/materialize/0.100.2/css/materialize.css">
  <script type="text/javascript" src="https://code.jquery.com/jquery-2.1.1.min.js"></script>
  <script src="https://cdnjs.cloudflare.com/ajax/libs/materialize/0.100.2/js/materialize.min.js"></script>
</head>
<body>
  <div id="navbar"></div>
  <div class="center-content page-container">

    <div class="progress" id="loader">
      <div class="indeterminate"></div>
    </div>
    <div style="position: relative" class="margin">
      <video onloadedmetadata="onPlay(this)" id="inputVideo" autoplay muted playsinline></video>
      <canvas id="overlay" />
    </div>

    <div class="row side-by-side">

      <!-- face_detector_selection_control -->
      <div id="face_detector_selection_control" class="row input-field" style="margin-right: 20px;">
        <select id="selectFaceDetector">
          <option value="ssd_mobilenetv1">SSD Mobilenet V1</option>
          <option value="tiny_face_detector">Tiny Face Detector</option>
        </select>
        <label>Select Face Detector</label>
```

Fig.5. webcamFaceExpression.html



Name	Date modified	Type
bbt.js	22-02-2022 22:06	JavaScript File
commons.js	05-05-2022 18:47	JavaScript File
faceDetectionControls.js	22-02-2022 22:06	JavaScript File
imageSelectionControls.js	22-02-2022 22:06	JavaScript File

Fig.6 json packadge



C:\Users\shreysaxena\face-api.js\examples\examples-browser\public\js\commons.js - Notepad++ [Administrator]
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
package.json
webcamFaceExpressionRecognition.html
commons.js
JavaScript file
length : 4,720 lines : 167
Ln : 1 Col : 1 Pos : 1
Windows (CR LF) UTF-8 INS
34°C Haze
ENG
19:41
05-05-2022

```
async function requestExternalImage(imageUrl) {
  const res = await fetch('fetch_external_image', {
    method: 'post',
    headers: {
      'content-type': 'application/json'
    },
    body: JSON.stringify({ imageUrl })
  })
  if (!(res.status < 400)) {
    console.error(res.status + ' : ' + await res.text())
    throw new Error('failed to fetch image from url: ' + imageUrl)
  }

  let blob
  try {
    blob = await res.blob()
    return await faceapi.bufferToImage(blob)
  } catch (e) {
    console.error('received blob:', blob)
    console.error('error:', e)
    throw new Error('failed to load image from url: ' + imageUrl)
  }
}

function renderNavBar(navbarId, exampleUri) {
  const examples = [
    {
      uri: 'face_detection',
      name: 'Face Detection'
    },
    {
      uri: 'face_landmark_detection',
```

Fig. 7 Common.js

5. RESULT

The complete result have 92% of accuracy .

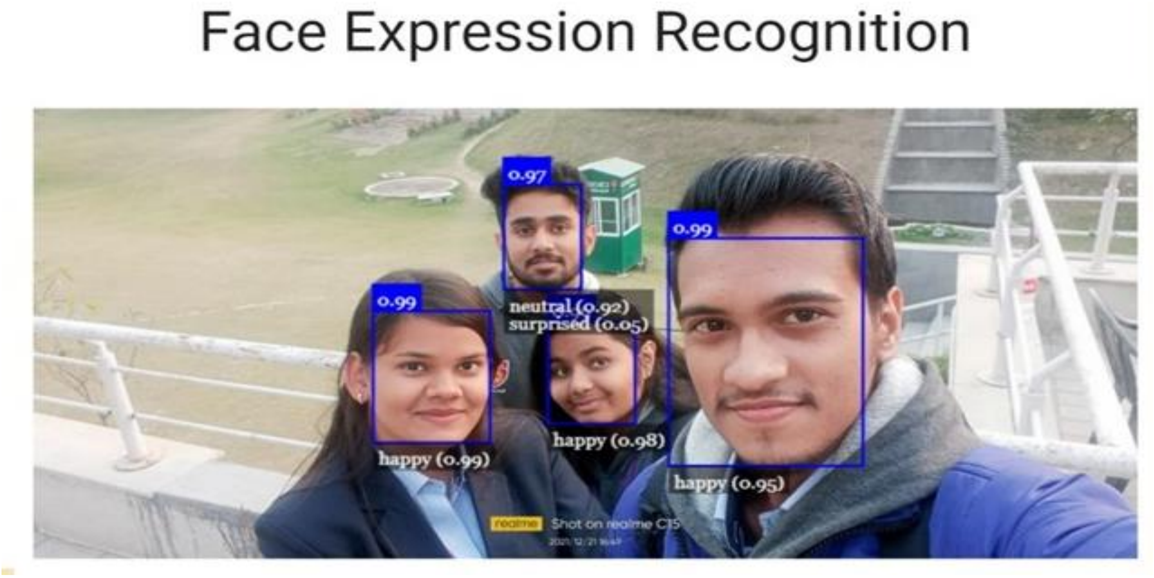


Fig 8. Emotion detection of 4-5 people

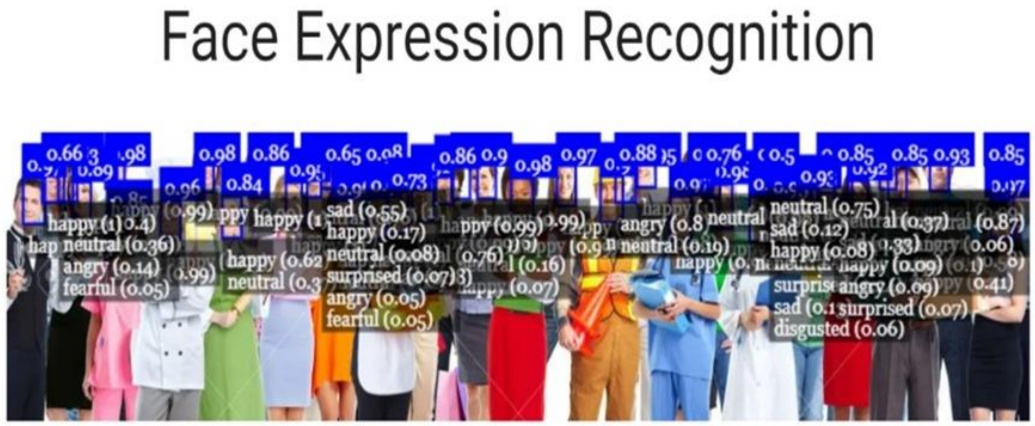


Fig 9. Emotion detection of 20 people

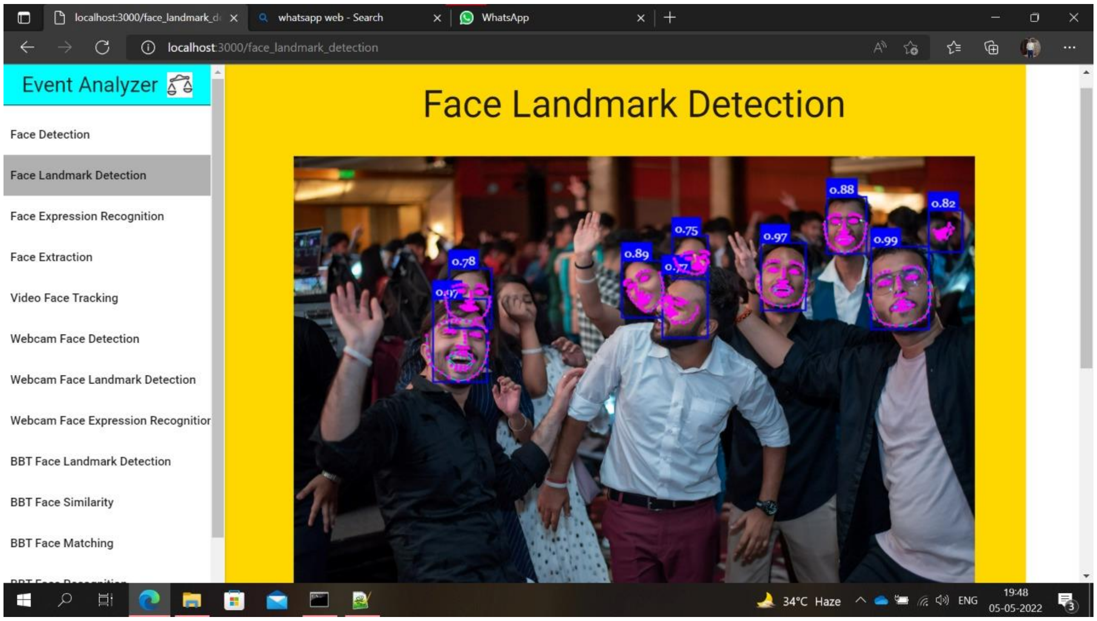


Fig 10. Feature extraction of group

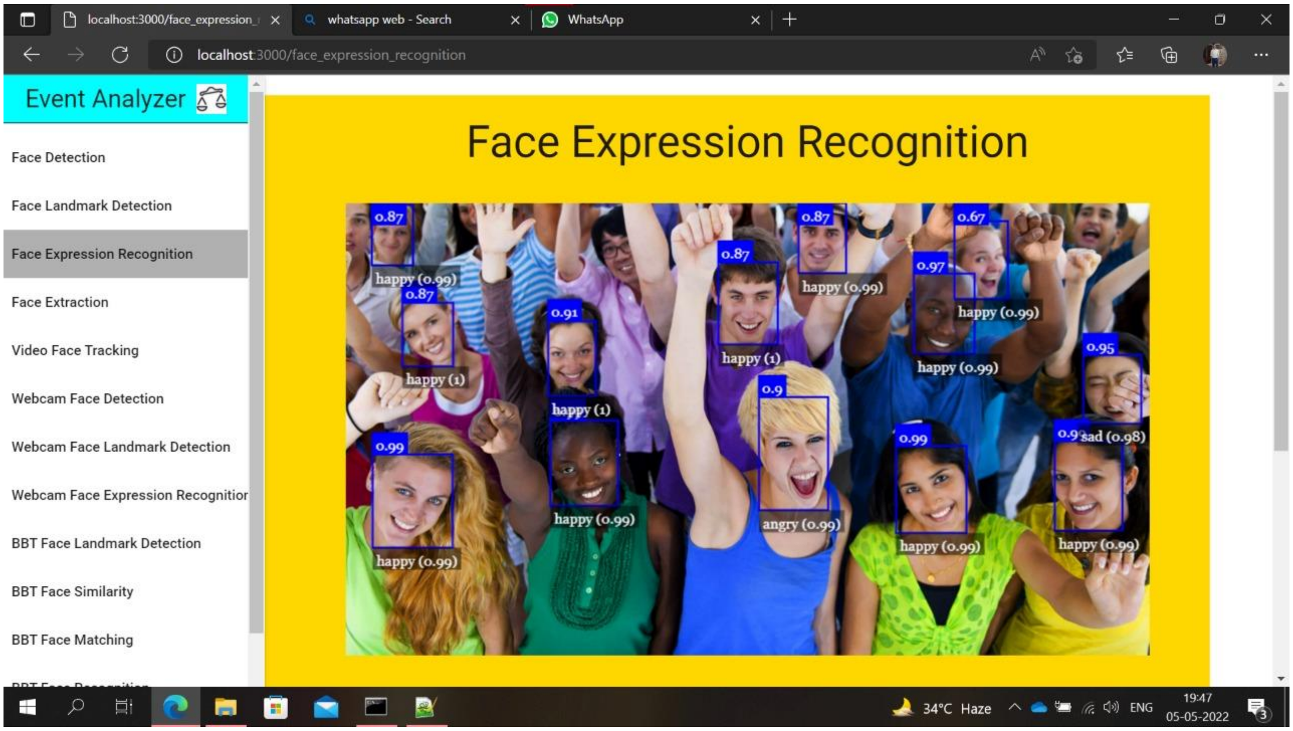


Fig 11. Emotion detection of 38 people

6. CONCLUSIONS

Nowadays, optimizers work on a stochastic basis Because of large datasets (millions of examples). However, this was true for a project. Given a Limited dataset, trying on the whole dataset could Have lead to a better feature learning. Also, the user of some optimizers reported on this Research would have had a different behavior. This Behavior can be displayed on the loss curve having A smoother shape or by avoiding an early Convergence. By having a larger dataset, networks with a larger Capability to learn features could be implemented. Thus, emotion classification could be achieved by Means of deep learning techniques. DL techniques will be able to solve this problem Given an enough amount of labelled examples. While feature engineering is not necessary, image Pre-processing boosts classification accuracy. Hence, it reduces noise on the input data.

7. FUTURE SCOPE

In future needs this application will automatically Detect the expression of the audience throughout The event and on basis of overall observation it will convey us the feedback of the entire event. It will majorly help us in business event and any Seminar.

8. REFERENCES

a. Searching for alignment in face recognition by Xiaqing Xu1, Qiang Meng1, Yunxiao Qin2, Jianzhu Guo3,4, Chenxu Zhao5*, Feng Zhou1, Zhen Lei.

b. A survey on deep learning for textual Emotion analysis in social networks SanchengPenga, Lihong Caob,∗, YongmeiZhouc, ZhouhaoOuyangd, AiminYangeXinguang Lia, WeijiaJiaf, Shui Yug

c. Emotion Detection and Analysis Using Machine Learning and Deep Learning Posted by Sylvester Kaczmarek in Science & Technology, Social & Psychology

d. Human Emotion Recognition System using Deep Learning Techniques (ijert.org) Emotion analysis of Arabic tweets using deep learning approach Massa Baali& Nada Ghneim

e. Face detection techniques: a review Ashu Kumar1 · Amandeep Kaur2 · Munish Kumar3

f. MediaPipe Hands: On-device Real-time Hand Tracking,(Fan Zhang, Valentin Bazarevsky, Andrey Vakunov, Andrei Tkachenka, George Sung, Chuo-Ling Chang, Matthias Grundmann)

g. Q. Le, T. Mikolov, Distributed representations of sentences and Documents, in: Proceedings of the 31st International Conference On Machine Learning, Beijing, China, 2014, pp. 1188–1196.

h. R. Kiros, Y. Zhu, R. R. Salakhutdinov, R. Zemel, R. Urtasun, A. Torralba, S. Fidler, Skip-thought vectors, Advances in neural Information processing systems (2015) 3294–3302.

i. B. Felbo, A. Mislove, A. Sogaard, I. Rahwan, S. Lehmann, Using Millions of emoji occurrences to learn any-domain representaTions for detecting sentiment, emotion and sarcasm, in: ProceedIngs of the 2017 Conference on Empirical Methods in Natural Language Processing, Copenhagen, Denmark, 2017, pp. 1616– 1626.

j. J. Devlin, M. Chang, K. Lee, K. Toutanova, Bert: Pre-training Of deep bidirectional transformers for language understanding, In: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: HuMan Language Technologies, Minneapolis, Minnesota, 2019, pp. 4171–4186.

k. A. Conneau, D. Kiela, H. Schwenk, L. Barrault, A. Bordes, SuPervised learning of universal sentence representations from natUral language inference data, in: Proceedings of the 2017 ConFerence on Empirical Methods in Natural Language Processing, Copenhagen, Denmark, 2017, pp. 670–680.

l. Y. Gao, Y. Xu, H. Huang, Q. Liu, L. Wei, L. Liu, Jointly learning Topics in sentence embedding for document summarization, IEEE Transactions on Knowledge and Data Engineering 32 (4) (2020) 688– 699.

m. Y. Hao, X. Liu, J. Wu, P. Lv, Exploiting Sentence Embedding For Medical Question Answering, in: Proceedings of the ThirtyThird AAAI Conference on Artificial Intelligence, 2019, pp. 938–945.

n. A. Radford, K. Narasimhan, T. Salimans, I. Sutskever, Improving Language understanding with unsupervised learning,technical Report (2018).

o. Age-unsupervised/language understanding paper.pdf

p. F. Hill, K. Cho, A. Korhonen, Learning distributed representaTions of sentences from unlabelled data, in: Proceedings of the 2016 Conference of the North American Chapter of the AssociaTion for Computational Linguistics: Human Language TechnoloGies, San Diego, California, 2016, pp. 1367– 1377.

q. Y. Sun, S. Wang, Y. Li, S. Feng, H. Tian, H. Wu, , H. Wang, ERNIE 2.0: A Continual PreTraining Framework for Language Understanding, in: Proceedings of the Thirty-Fourth AAAI ConFerence on Artificial Intelligence (AAAI 2020), 2020, pp. 8968– 8975.

r. S. Subramanian, A. Trischler, Y. Bengio, C. J. Pal, Learning genEral purpose distributed sentence representations via large scale Multi-task learning, in: Proceedings of the Sixth International Conference on Learning Representations, Vancouver, Canada, 2018, pp. 1–16.

s. D. Cer, Y. Yang, S. Kong, N. Hua, N. Limtiaco, R. S. John, N. Constant, M. Guajardo-Cespedes, S. Yuan, C. Tar, Y. Sung, B. Strope, R. Kurzweil, Universal sentence encoder for English, In: Proceedings of the 2018 Conference on Empirical Methods In Natural Language Processing (System Demonstrations), BrusSels, Belgium, 2018, pp. 169–174.

t. M. Pagliardini, P. Gupta, M. Jaggi, Unsupervised learning of senTence embeddings using compositional n-gram features, in: ProCeedings of NAACL-HLT 2018, New Orleans, Louisiana, 2019