CERTIFICATE

Certified that minor project work entitled “EMOTION TRACKER” is a bonafide work carried out in the 5 th semester by “Arpit pundir,Piyush gupta,Gaur jetharam” in partial fulfillment for the award of Bachelor of Computer Science and Engineering from jaypee Institute of Information Technology Noida during the academic year 2019-2020.

Project Guide **DepartmentCoordinator,CSE**

ACKNOWLEDGEMENT

First and foremost, we would like to thank Head of the Department of Computer Science and Engineering, Jaypee Institute of Information Technology who gave the opportunity to involve in such work of producing something new by our own. And thanks a lot to **Ms. Aditi Sharma** ma'am who guided us through doing this project, provided with invaluable advice, helped us in difficult periods and provided equipment for our projects. Her willingness to motivate us contributed tremendously to the success of these projects. Besides, we would like to thank all CSE staff members who helped us by giving advice and providing equipment which we needed.

Arpit Pundir
Piyush Gupta
Gaur Jetharam

Introduction

It was in 2013 when the research for facial expression recognition started. Kaggle announced facial expression recognition challenge in 2013. Researchers are expected to create models to detect 7 different emotions from human being faces. However, recent studies are far away from the excellent results even today. That's why this topic is still satisfying subject.

That time a dataset is generated which was named as FER2013. Both training and evaluation operations would be handled with FER2013 dataset. Compressed version of the dataset takes 92 MB space whereas uncompressed version takes 295 MB space. There are 28K training and 3K testing images in the dataset. Each image was stored as 48×48 pixel. The pure dataset consists of image pixels (48×48=2304 values) and emotion of each image. Emotions are as: Disgust, Anger, Fear, Sadness, Happiness, Surprise and Contempt.

Facial Expression Recognition is an emerging field comes in the domain of image processing and machine learning. It has several applications in various areas. Fully automatic facial expression system which works on real time is very much helpful in interpreting different facial gestures which is used in a wide variety of applications like human computer interaction, behavioural research and vision systems.

Problem Statement or Application focused:

Real time facial expressions are helpful in the field of robotics and there are several other areas like Animations, Psychiatry, Video Games, Sensitive Music, Educational Software, Forensis, Medical Science, Criminal Interview etc. But here we focused on a different application where this technology is going to help in the educational field.

Age of 3-7 is the period of keeping kids under observations to see how they are growing or developing. Because kids can’t tell their feelings to others so we have to build such an environment by which we can understand the feelings of kids. As the behaviour and activities of child plays an important role in deciding the future characteristics and personality. No one more than parents can take care of the child or look after him/her. But as kids spend more time in school rather than at home. So, it is not possible to look after the kid all the time. As we clarified that facial expressions helps in interpersonal communications. So Facial Expression Recognition can be used to resolve this problem. Schools should have the facility of keeping children under surveillance and by this we can extract the information of whole day expressions and see what is the reaction of individual kid. And daily report should be sent to their home so that their parents get to know about their child's situation. Here Facial Expression recognition comes in role.

Related Work:

Application we are working on is not focused or used till now as we researched about it after we got the idea. But the feature of face recognition and facial expression is trending and too much focused field of supervised learning. Some of the research we have focused or studied to learn this are described below.

References	Preprocessing	Feature Extraction	Classification	Post-Processing	Recognition Performance
Bourel et al.	Tracking of facial landmarks using a point tracker specialised to work on facial features.	Scalar quantisation of facial dynamics (which are represented as temporal evolutions of scale-normalised geometric measurements between facial landmarks).	Classifier: rank weighted k-nearest neighbour classifier. Classes: 6 basic expression prototypes (anger, disgust, fear, joy, sadness and surprise).		Data set: 300 image sequences, front face view, many subjects (database: CMU-Pittsburgh AU-Coded Face Expression Image Database). Recognition accuracy: relatively little degradation in recognition under partial face occlusion or tracker noise.
Pantic and Rothkrantz	Location / tracking of the face and its parts (front and profile	Extraction of static geometric measurements from landmarks on	Classifier: rule-based expert system for each stage of a two-stage	Two-stage Classification Architecture	Data set: 265 still images (two views each, front and profile), more than 8

	contours of the head, eyebrow region, eye region, mouth region) using multiple detectors (e.g. snakes, neural networks,	the contours of eyebrows, eyes, nostrils, mouth, and chin.	classification hierarchy: encoding of face geometry into AUs, followed by classification into basic expression prototypes. Classes: 28 AUS and 6 basic expression prototypes (happiness, sadness, surprise, anger, fear, and disgust).		subjects. Recognition accuracy: 91% recognition of basic expression prototypes.
Essa and Pentland	Location of the face by fitting a 3D mesh model of face geometry to a 2D face image. The model fitting process is based on View-based and Modular Eigenspace methods, coupled to image warping Face tracking by the mesh model is tied to optical flow	Alternative features: (i) peak activation of each muscle. Muscle activation is extracted using a physics-based model of facial deformation. The deformation is estimated from the optical flow. (ii) 2D motion-energy computed from motion estimates, which have been	Classifier: alternative similarity measures (i) maximum correlation with muscle aäivation template, (ii) minimum distance to motion energy template. Classes: 5 facial expressions (smile, surprise, anger, disgust, raised brows).		Data set: 52 image sequences, front face view, 8 subjects. Recognition accuracy: 98% recognition of facial expressions.

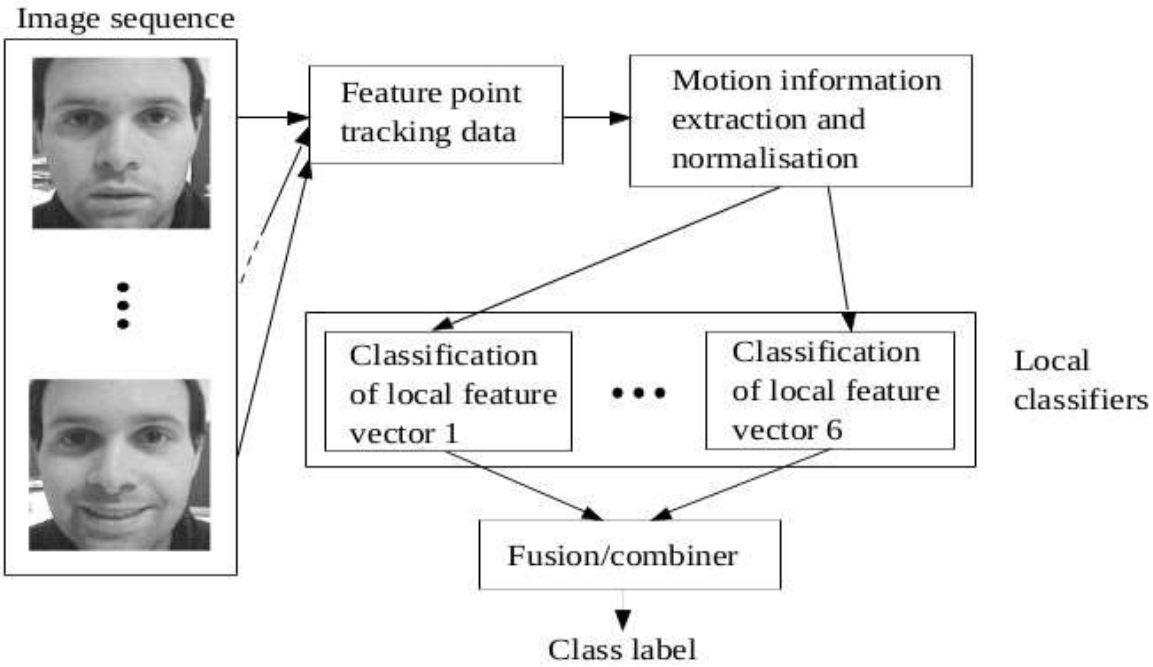
	computation.	corrected by the physics-based model.			
Tian et al.	Segmentation and tracking of permanent face parts and transient features based on: Gaussian mixture model (for lip colour), deformable template (for lip shape, and eyes), Lucas- Kanade tracking algorithm (for brows and cheeks), Canny edge detector (for wrinkles and furrows).	Continuous and discrete (state-based) representation of magnitude and direction for motion of face parts (lips, eyes, brows, cheeks), as well as state-based (present / absent) representation of transient features (furrows and wrinkles). The continuous representation is normalised against image scale and in plane head motion.	Classifiers: 1 multilayer perceptron for upper-face AUs, 1 multi-layer perceptron for lower-face Aus. Classes: 6 upper-face AUs, 10 lower-face AUs, and neutral expression. (Classified into a single AU class or a combination of AU classes).		(i) Database: CMUPittsburgh AU-Coded Face Expression Image Database. Recognition accuracy: 96.4% recognition of upperface AUs, and neutral expression. Data set: 50 image sequences, front face view, 14 subjects. 96.7% recognition of lower-face AUs, and neutral expression. Data set: 63 image sequences, front face view, 32 subjects. (ii) Database: Ekman-Hager Facial Action Exemplars Database. Recognition accuracy: 93.3% recognition of upperface AUs, lower-face AUs, and neutral

					expression. Data set: 122 image sequences, front face view, 21 subjects.
Lien et al.	Alternative segmentation or motion estimation processes: facial feature point tracking, dense optical flow, or edge detection. Geometric normalisation against rigid translation, rotation, and scaling of the head. Segmentation: cropping to region of interest (upper or lower face).	Alternative representations: (i) displacement (of 6 points the on upper boundary of brows) relative to the first frame (ii) PCA of dense optical flow (iii) block-based density and variance of pixels, which show significant change in edge magnitude relative to the first frame.	Classifier: vector quantiser followed by hidden Markov models. Classes: 3 upper-face AUS associated with brow movement, and neutral expression. (Classified into a single AU class or a combination of AU classes).		Data set: 260 image sequences, front face view, 60 subjects. Recognition accuracy: 85% AU recognition (featurepoint displacement, or edge density and variance). 93% AU recognition (PCA features of dense optical flow).

1. Paper by Fabrice Bourel:

They present a new approach for the recognition of facial expressions from video sequences in the presence of occlusion. Although promising results have been reported in the literature on automatic recognition of facial expressions, most techniques have been assessed using experiments performed in controlled laboratory conditions which do not reflect real-world conditions. The goal of the work presented herein, is to develop recognition techniques that will overcome some limitations of current techniques, such as their sensitivity to partial occlusion of the face.

Architecture of Recognition System:



Proposed Work:

Our basic idea was to make a platform for school as well as parents where they can make surveillance on the emotional condition of the child as the kid can't share the inner feelings because they also don't know what's going on in their life but parents can understand the child by watching his/her activities which can't be possible in this hectic life schedule.

So, we want something where we can aggregate the whole emotional behaviour of the child of all the time under surveillance to which parents have access to go through all the data.

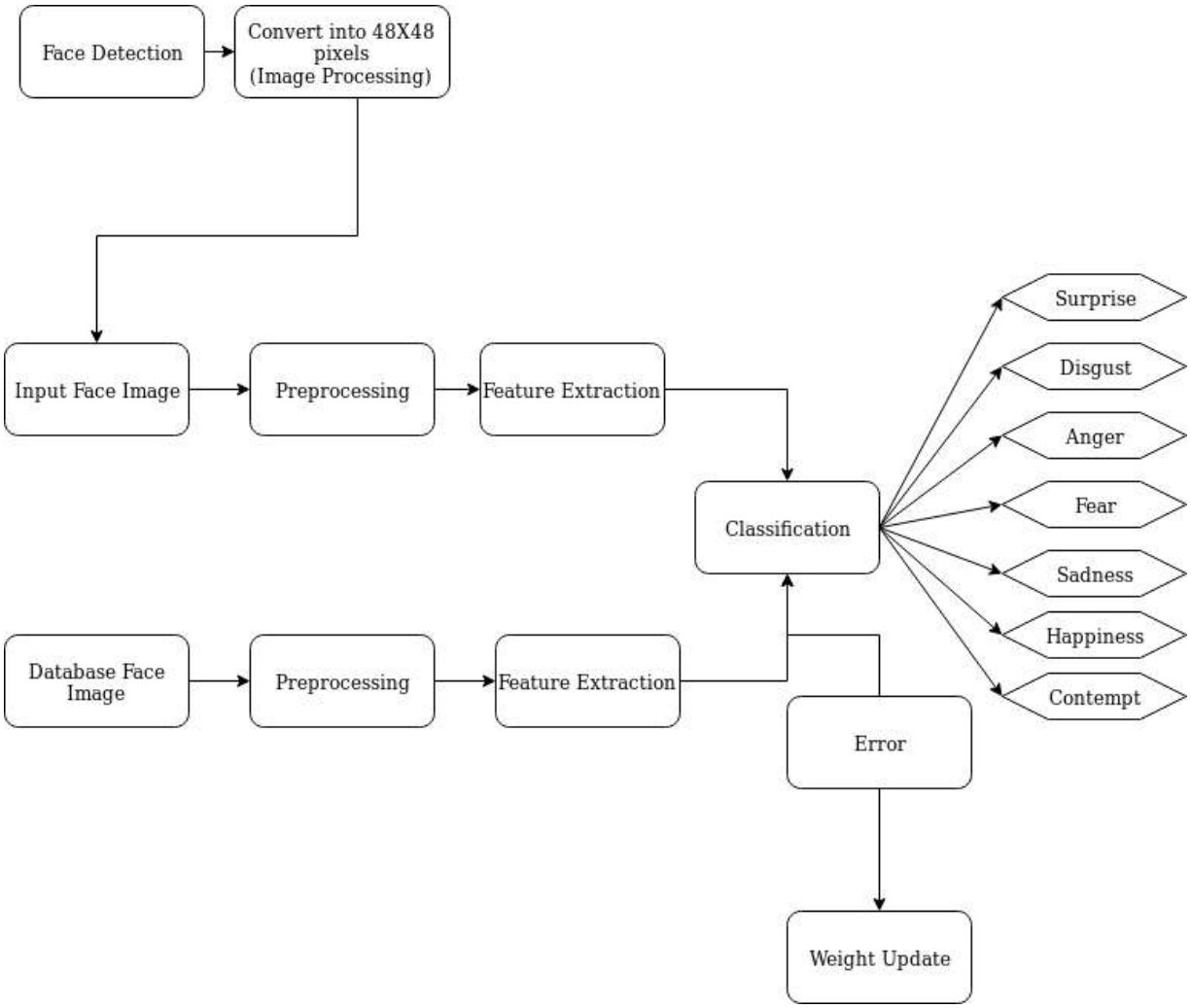
Here we come up with the idea of implementing such platform with the use of machine learning from which we are extracting the face data and its emotions to maintain the track of the emotion corresponding to its identity.

We divided this project into two parts where one part is for implementing machine learning on the person's face using face recognition and facial expression recognition neural networks. And after collecting the data we have to show this data in a structured form to the client (in this case client can be parents). For this purpose we designed a website using MERN stack technology.

Now below is a brief description of both the faces of our project.

Face Identification and Facial Expression Recognition:

Here is the flow diagram of the whole process we are working on.



The main and most important part of our project are neural networks which we used for face identification and facial expression recognition. Below is the short explanation of how neural networks work.

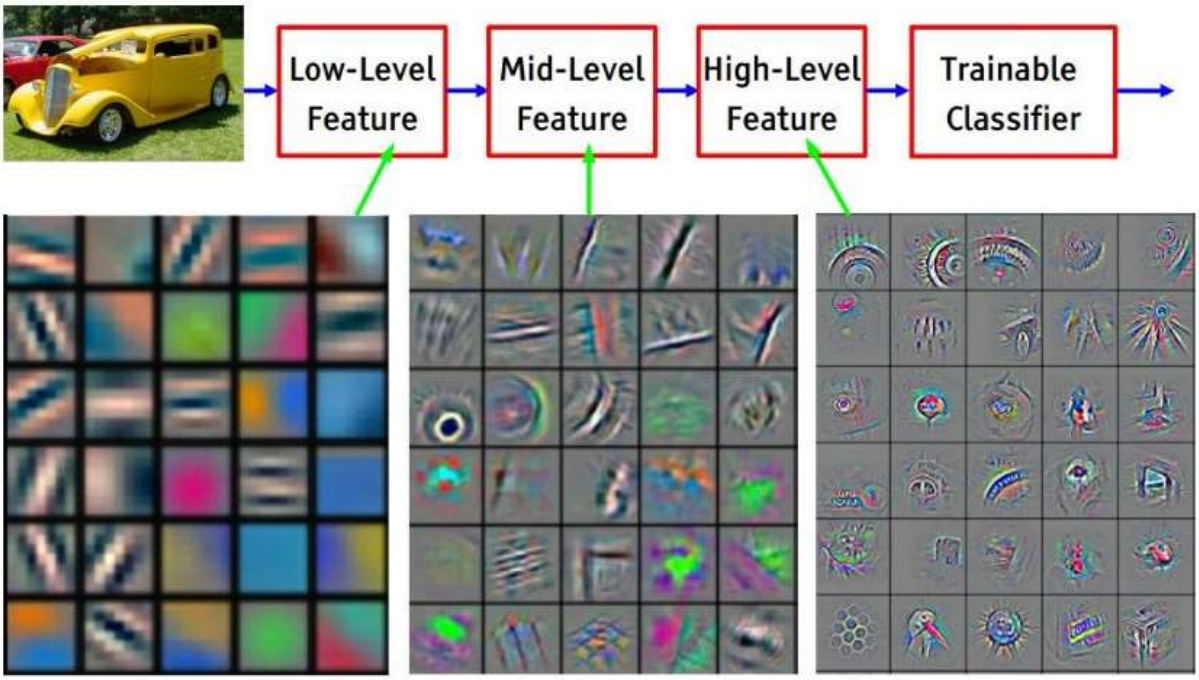
How Neural Network Works:

We have to design a model with an input layer and an output layer including some hidden layers which help in feature extraction.

What do the hidden layers learn??

The hidden layers in a CNN are generally convolution and pooling(downsampling) layers. In each convolution layer, we take a filter of a small size and move that filter across the image and perform convolution operations. Convolution operations are nothing but element-wise matrix multiplication between the filter values and the pixels in the image and the resultant values are summed.

The filter's values are tuned through the iterative process of training and after a neural net has trained for certain number of epochs, these filters start to look out for various features in the image. Take the example of face detection using a convolutional neural network. The earlier layers of the network looks for simple features such as edges at different orientations etc. As we progress through the network, the layers start detecting more complex features and when you look at the features detected by the final layers, they almost look like a face.



Now, let's move on to pooling layers. Pooling layers are used to downsample the image. The image would contain a lot of pixel values and it is typically easy for the network to learn the features if the image size is progressively reduced. Pooling layers help in reducing the number of parameters required and hence, this reduces the computation required. Pooling also helps in avoiding overfitting. There are two types of pooling operation that could be done:

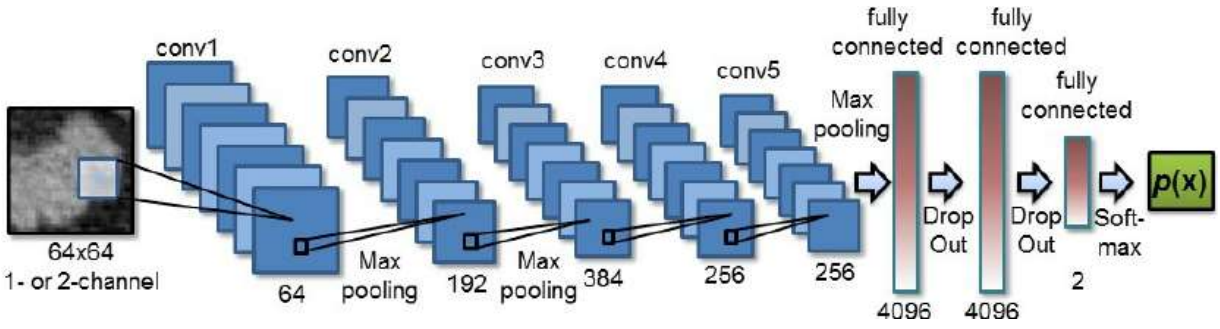
- Max Pooling — Selecting the maximum value
- Average Pooling — Sum all of the values and dividing it by the total number of values

Average pooling is rarely used, you could find max pooling used in most of the examples.

Why CNN(Convolutional Neural Networks)?

In fact, machine learning engineerArden Dertat in an article inTowards Data Science states that CNN is the most popular deep learning model. According to Dertat, the recent surge of interest in deep learning is thanks to the effectiveness and popularity of convnets. Such is the accuracy that CNNs have become the go-to models for a lot of industry applications. For example, they are used for recommender systems, natural language processing and more.

The main advantage of CNN compared to its predecessors is that it automatically detects the important features without any human supervision.In fact, machine learning engineerArden Dertat in an article inTowards Data Science states that CNN is the most popular deep learning model. According to Dertat, the recent surge of interest in deep learning is thanks to the effectiveness and popularity of convnets. Such is the accuracy that CNNs have become the go-to models for a lot of industry applications. For example, they are used for recommender systems, natural language processing and more. The main advantage of CNN compared to its predecessors is that it automatically detects the important features without any human supervision.



In terms of architecture, the key building block of CNN is the convolutional layer. According to a MathWork post, a CNN convolves learned features with input data, and uses 2D convolutional layers, making this architecture well suited to processing 2D data, such as images. Since CNNs eliminate the need for manual feature extraction, one doesn't need to select features required to classify the images.

How CNN work is by extracting features directly from images and the key features are not pretrained; they are learned while the network trains on a collection of images, the post notes. It is the automated feature extraction that makes CNNs highly suited for and accurate for computer vision tasks such as object/image classification.

Emotion Tracker Web App

Intro

Emotion Tracker Web App is a support application for Emotion Tracker Software. Users interact with the Emotion Tracker Software only through the web app. The web app can be treated as a utility part of the project.

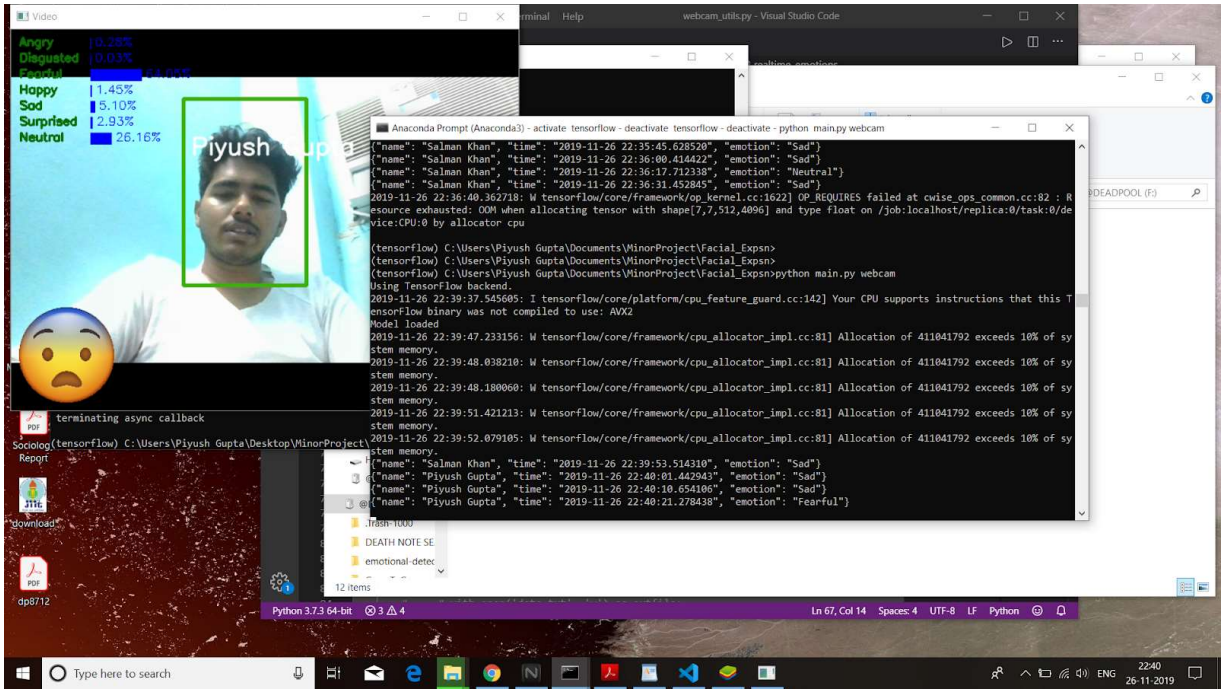
The web app takes data from the Emotion Tracker through an API and stores the data in the respective user accounts hosted on MongoDB compass and atlas. The user can then signup and login through React.js based client side of the project and get information about his/her tracked data.

Data comes in the form of cards each time the user was tracked, and each card shows the percentage of the respective emotion at the moment the user was tracked.

WorkFlow

Step1

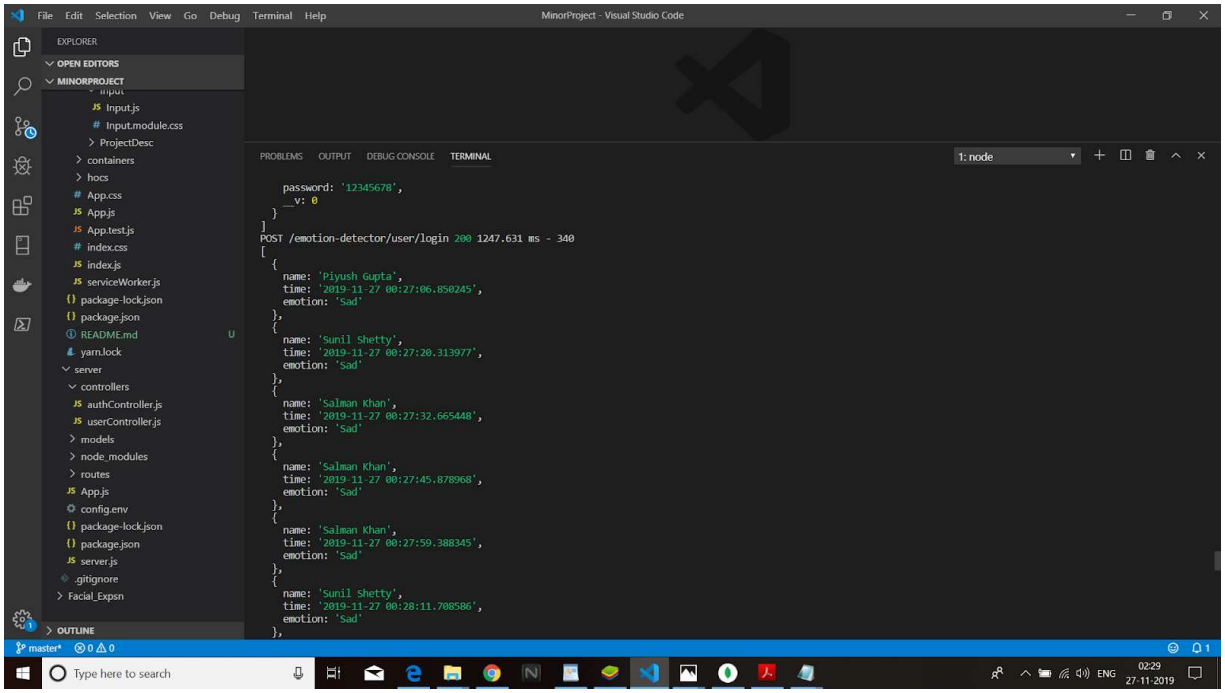
In the first step, an image detector detects the person and run the models and identifies the person as per the accuracy.



Ans sends the data to the server along with all the results.

Step2

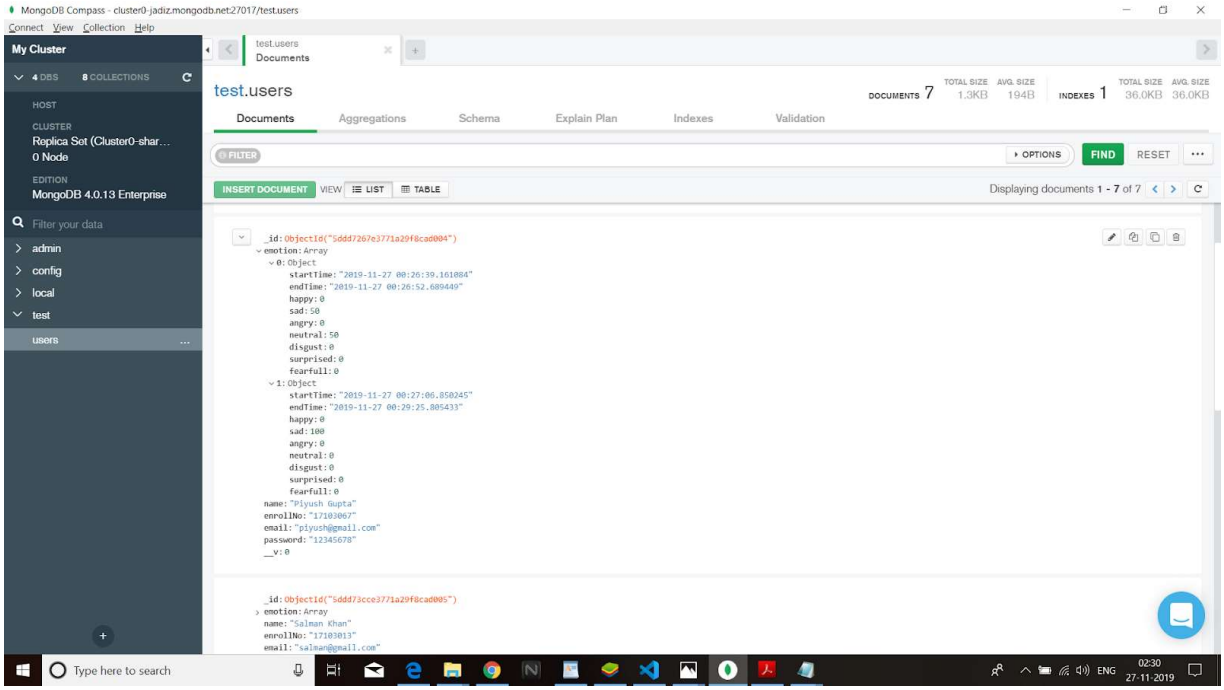
In step 2, req strikes at the server and server process the data and collects all the instances in which the user has been scanned and calculates the final values of the emotions.




```
exports.data = async(req, res, next)=>{
  try{
    const data = req.body.data;
    const name = data[0].name
    //console.log(req.body)
    //console.log(data)
    const id = data[0].id
    let len = data.length
    const startTime = data[0].time
    const endTime = data[len-1].time
    let happy = 0, neutral = 0, angry = 0, sad = 0, fearfull = 0, disgust = 0, surprised = 0
    console.log(happy, surprised)
    data.forEach(element => {
      console.log(element)
      if(element.emotion == "Happy"){
        happy+=1
      }else if(element.emotion == "Angry"){
        angry+=1
      }else if(element.emotion == "Sad"){
        sad+=1
      }else if(element.emotion == "Neutral"){
        neutral+=1
      }else if(element.emotion == "Fearfull"){
        fearfull+=1
      }else if(element.emotion == "Disgust"){
        disgust+=1
      }else{
        surprised+=1
      }
    })
```

Step 3

In the third step, data is sent finally to the MongoDB database, on the compass as well as the atlas.



Tech Stack

1. **MongoDB**
    a. **MongoDB Compass**
    b. **MongoDB Atlas**
2. **Express**
3. **React.js**
4. **Node.js**

Why MongoDB?

Organizations of all sizes are adopting MongoDB because it enables them to build applications faster, handle highly diverse data types, and manage applications more efficiently at scale.

Development is simplified as MongoDB documents map naturally to modern, object-oriented programming languages. Using MongoDB removes the complex object-relational mapping (ORM) layer that translates objects in code to relational tables. MongoDB's flexible data model also means that your database schema can evolve with business requirements.

MongoDB can also be scaled within and across multiple distributed data centers, providing new levels of availability and scalability previously unachievable with relational databases like MySQL. As your deployments grow in terms of data volume and throughput, MongoDB scales easily with no downtime, and without changing your application. In contrast, achieving scale with MySQL often requires significant custom engineering work.

Developer Productivity with JSON Documents

Working with data as flexible JSON documents, rather than as rigid rows and columns, is proven to help developers move faster. It's not hard to find teams who have been able to accelerate development cycles by 3-5x after moving to MongoDB from relational databases. Why is this?

Documents are natural. Documents represent data in the same way that applications do. Unlike the tabular rows and columns of a relational database, data can be structured with arrays and subdocuments – in the same way, applications represent data, like lists and members/instance variables respectively. This makes it much simpler and faster for developers to model how data in the application will map to data stored in the database.

Documents are flexible. Each document can store data with different attributes from other documents. As an example, consider a product catalog where a document storing details for an item of mens' clothing will store different attributes from a document storing details of a tablet. This is a property commonly called "polymorphism". With JSON documents, we can add new attributes when we need to, without having to alter a centralized database schema. At worst, this causes downtime, at best, significant performance overhead in a relational database. The flexibility documents bring allows the developer to more easily handle the semi and unstructured data generated by modern mobile, web, and IoT applications.

Documents make applications fast. With data for an entity stored in a single document, rather than spread across multiple relational tables, the database only needs to read and write to a single place. Having all the data for an object in one place also makes it easier for developers to understand and optimize query performance.

```
const userScheema = mongoose.Schema({
  name:{
    type: String,
    required: true,
  },
  email:{
    type: String,
    required: true,
    validate: [validator.isEmail, "Please provide a valid email address"]
  },
  enrollNo:{
    type: String,
    required: true,
    validate: [validator.isNumeric, "Please provide a valid enroll num"]
  },
  password:{
    type: String,
    required: true,
    minlength: 8
  },
  emotion:{
    type: Array
  }
})
```

It's for these reasons that MySQL, and other relational databases, have added support for JSON. However, simply adding a JSON data type does not bring the developer productivity benefits of a document database to MySQL. Why? Because MySQL's approach can detract from developer productivity, rather than improve it. Consider the following:

Proprietary Extensions: Querying and manipulating the contents of a JSON document requires the use of separate MySQL-specific SQL functions to access values, which will not be familiar to most developers. In addition, they are not supported or recognized by 3rd party SQL tools, such as BI platforms, data warehouse connectors, ETL and ESB pipelines, and more.

How is MongoDB different: The MongoDB API is widely understood, and adopted by industry-standard tools and connectors. Several mega-vendor database companies have even adopted the MongoDB API themselves.
Legacy Relational Overhead: Even with JSON support, MySQL users are still tied to multiple layers of SQL/relational functionality to interact with JSON data – low-level JDBC/ODBC drivers and Object Relational Mappers (ORMs). To experienced MySQL developers, these layers may be familiar, but to many other developers who want to interact with documents and data through APIs that are natural and idiomatic to their programming language, these layers impose high learning overhead. ORMs are also generally recognized as hard to optimize for performance and query efficiency – even for experienced relational developers. In addition, query optimization statistics for JSON data are more limited than those maintained for regular relational data types.

MongoDB drivers are implemented in the methods and functions that are idiomatic and natural to the programming languages used by developers.
Complex Data Handling: When using JSON data, MySQL drivers do not have the capability to properly and precisely convert JSON into a useful native data type used by the application. This includes different types of numeric values (e.g. floating points, 64-bit integers, decimals) timestamps, and dates, or a Map or List in Java or a Dictionary or List in Python. Instead, developers have to manually convert text-based JSON in their application, losing the ability to have fields that can take on multiple data types in different documents (polymorphism) and making the computation, sorting, and comparison of values difficult and error-prone.

How is MongoDB different: Binary Encoded JSON (BSON) used by MongoDB and its drivers support advanced data types not supported by regular text-based JSON.
No Data Governance: MySQL offers no native mechanism to validate the schema of JSON inserted or updated in the database, so developers need to add either application or database-side functionality to apply governance controls against the data.

How is MongoDB different: Schema validation, based on the JSON Schema IETF standard, allows developers and DBAs to define and enforce a prescribed schema structure for each MongoDB collection.
Schema Rigidity: MySQL users still need to define a schema for their regular relational data. If the schema is then modified to accommodate new application requirements, the table is locked for some operations until existing data is copied into the new schema, requiring applications to be quiesced during schema migration.

How is MongoDB different: Developers and DBAs can combine the flexibility of a fully dynamic schema with the governance controls needed for some applications across all data stored in the database, not just subsets of it.

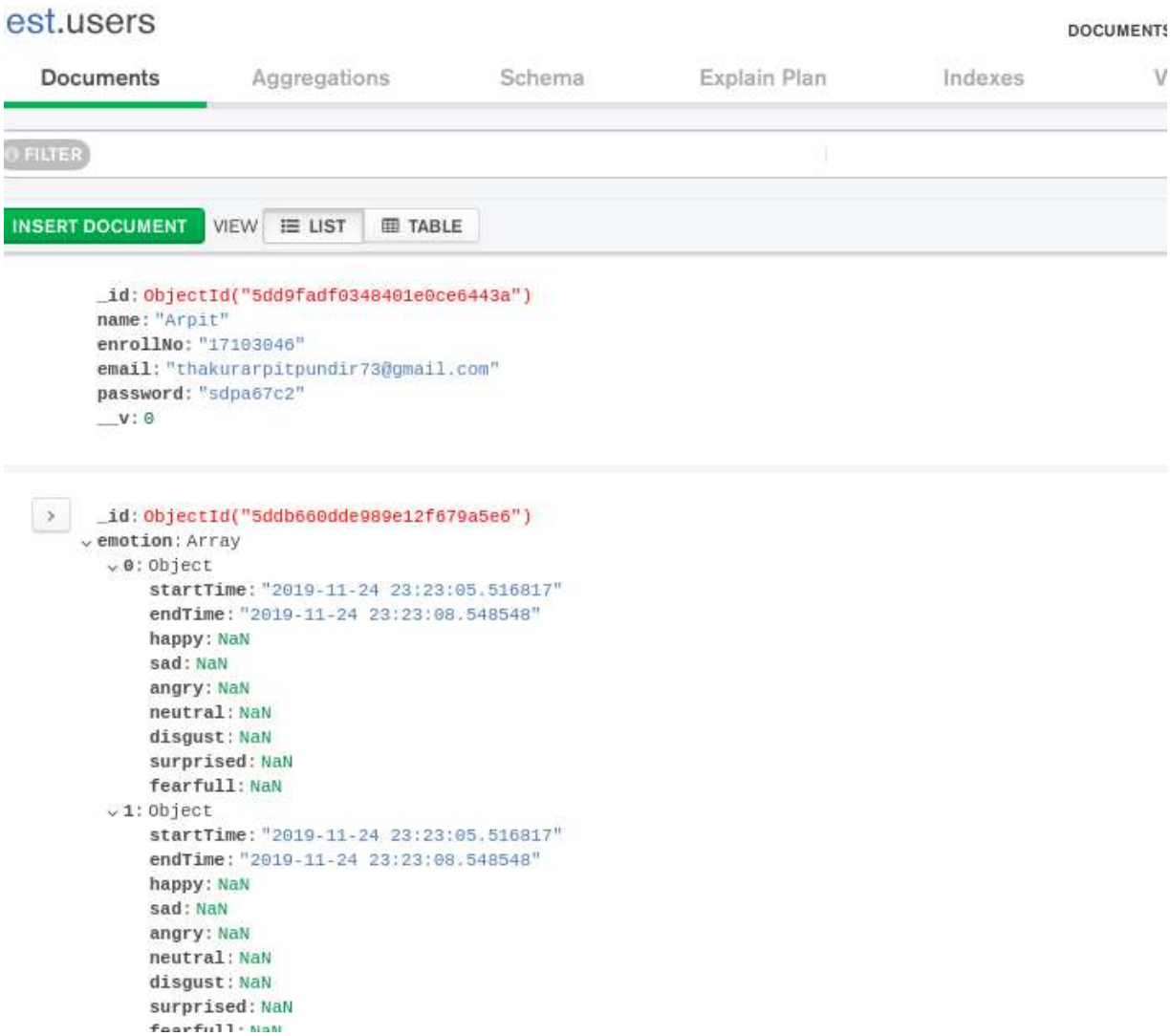
What is MongoDB Compass

MongoDB Compass is the GUI for MongoDB. Compass allows you to analyze and understand the contents of your data without formal knowledge of MongoDB query syntax. In addition to exploring your data in a visual environment, you can also use Compass to optimize query performance, manage indexes, and implement document validation.

Why MongoDB Compass

Know your data with built-in schema visualization

MongoDB Compass analyzes your documents and displays rich structures within your collections through an intuitive GUI. It allows you to quickly visualize and explore your schema to understand the frequency, types, and ranges of fields in your data set.



->Get immediate insight into server status and query performance

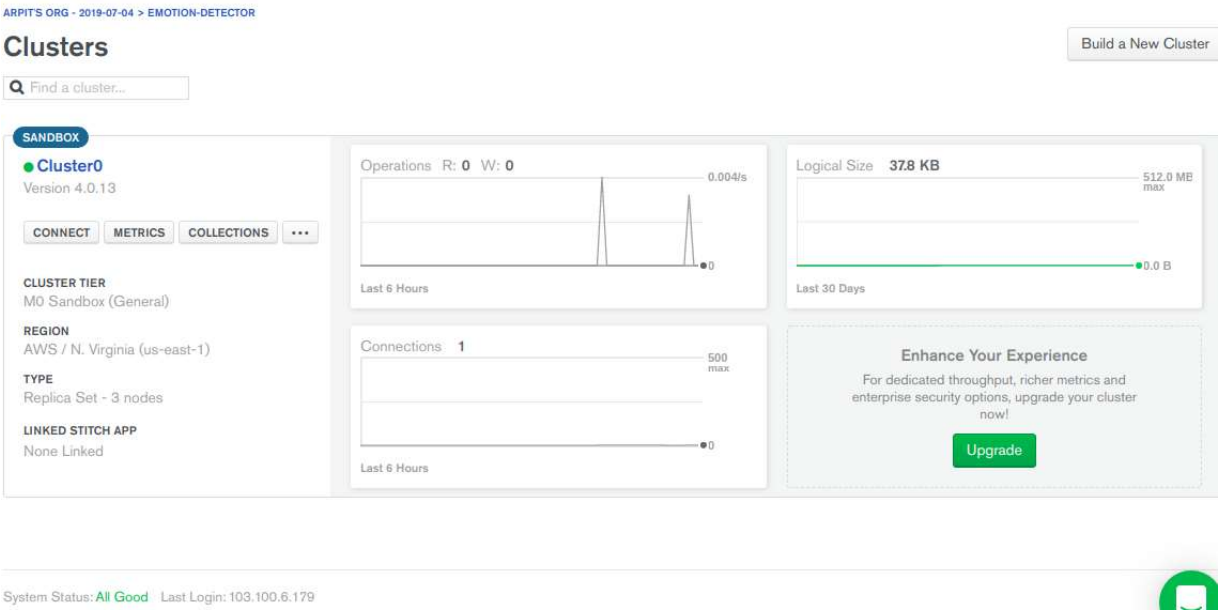
Real-time server statistics let you view key server metrics and database operations. Drill down into database operations easily and understand your most active collections.

->Visualize, understand, and work with your geospatial data
Point and click to construct sophisticated queries, execute them with the push of a button and Compass will display your results both graphically and assets of JSON documents.

->A better approach to CRUD makes it easier to interact with your data
Modify existing documents with greater confidence using the intuitive visual editor, or insert new documents and clone or delete existing ones in just a few clicks.

MongoDB Atlas

MongoDB Atlas is the global cloud database service for modern applications. Deploy fully managed MongoDB across AWS, Azure, or GCP. Best-in-class automation and proven practices guarantee availability, scalability, and compliance with the most demanding data security and privacy standards. Use MongoDB's robust ecosystem of drivers, integrations, and tools to build faster and spend less time managing your database.



Express

ExpressJS is a prebuilt NodeJS framework that can help you in creating server-side web applications faster and smarter. Simplicity, minimalism, flexibility, scalability are some of its characteristics and since it is made in NodeJS itself, it inherited its performance as well.

In short, ExpressJS did for NodeJS what Bootstrap did for HTML/CSS and responsive web design.

It made coding in NodeJS a piece of cake and gave programmers some additional features to extend their server-side coding. ExpressJS is hands down the most famous NodeJS framework- so much so that when most people talk about NodeJS they surely mean NodeJS+ExpressJS.

Rapid Server-Side Programming

Being a Node.js framework, Express.js packages many of the commonly used Node.js features, into functions that can be easily called anywhere on the program. As a result, complex tasks that would otherwise take a Node developer several hundred lines and several hours to program can easily be done by Express JS developers in just a few lines of code and within a few minutes. Express web application development is therefore much quicker than pure Node.js development.

Routing

Routing allows a web application to preserve web page states through their URLs. These URLs may be shared with other users, and visiting these URLs will take users to the exact page state that was originally shared. Node.js has a routing mechanism, but it's a basic and rudimentary one. Express.js offers a more advanced and efficient routing mechanism that is able to handle highly dynamic URLs.

Debugging

All developers encounter bugs with every project that can cause entire applications to malfunction, and one of the most critical tasks of developers is identifying the source of these bugs and correcting them in the quickest possible time. Fortunately, Express.js provides an easy debugging mechanism to allow developers to quickly pinpoint which part of the application causes bugs.

Templating

Express.js provides a templating engine that allows web pages to have dynamic content by constructing HTML templates on the server-side, replacing dynamic content with their proper values, and then sending these to the client-side for rendering. In addition to enabling dynamic content, it also takes a significant load from the client side which may have highly variable hardware specifications, and as such, it can make applications more efficient.

Middleware

Express.js uses middleware to systematically arrange different function calls. A middleware is a chunk or cluster of code that has access to a user's request, the application's response, and the next middleware to be used. With such an architecture, it becomes easy for Express js developers to add, remove, or modify various features to and from the application, giving high scalability to the application.

React.js

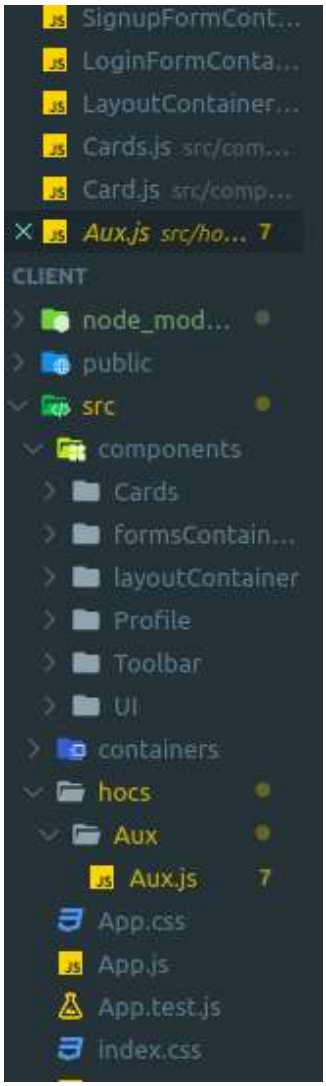
Client-Side code of the project is fully based on the React.js. Team has split the code base fully into the components so that the repetition of the code is minimum and the component which are being used multiple times should not be written more than once and we can simply use make an instance of the component code we have written.

Virtual DOM

Considered the next biggest leap in web development since AJAX, the virtual DOM (short for Document Object Model) is the core reason why React enables the creation of fast, scalable web apps. Through React's memory reconciliation algorithm, the library constructs a representation of the page in virtual memory, where it performs the necessary updates before rendering the final web-page into the browser.

Notable Boost In Performance

The Virtual DOM mentioned earlier emphatically increases the speed of modern web applications because it eliminates the usage of code-heavy frameworks such as Jquery and other bootstrapping libraries. React itself is sufficient in creating awesome looking front-end designs and combined with its super-fast rendering capabilities is a natural fit for companies to utilize it in their services.



Seamless SEO Integration

For any online business, content is king. Search Engine Optimization is the gateway to boost user traffic onto its platform. React significantly reduces page load time through faster rendering speed, adapts it performance in real-time based on current user traffic, features that are otherwise not perfectly handled by most frameworks.
This particular aspect is essential for the success of businesses because faster speed is directly proportional to more users which converts to the main goal of companies: Revenue. This is also emphasized by Moz in their article that effective use of SEO will improve the app's ranking on Google search and reaching the number one spot is the holy grail of every web-based platform.

Ease of Migration

When engineers and managers collaboratively decide upon migrating from an older technological infrastructure to a new one, certain questions arise on the level of effort and time required in performing the task. Google's Angular framework; for instance, faced a very alarming compatibility issue between its first-generation Angular 1.x and the futuristic Angular 2. Companies intent on shifting to the newer version had to invest time to train developers in the latest technology.

React; on the other hand, is lightweight and practically wrapped around the same JavaScript standards that made developers and managers alike, to fall in love with this language in the first place. React code can be added anywhere onto existing infrastructure without the worry of shutting down the system for maintenance and also less dependency to reinvent the wheel.

Fuse technologies for bigger Impact

React makes use of the ever-popular HTML and Javascript frameworks by integrating them together. Extend the two technologies onto CSS facilitates the designing of advanced looking web-interfaces. React is a very API friendly library and is extensible across a multitude of frameworks. It works seamlessly with Google's ionic Material Design and SemanticUI frameworks to leverage user interface development.

Quickly Debug Faults

Facebook has created a dedicated debugging mechanism to isolate UI errors and bugs to the exact component causing it. The browser itself spits out relevant data about the erroneous line of code, web page and/or section to make the required correction, making the lives of developers a whole lot easier.

Improve Code Stability With Tests

Businesses looking to create a fault-tolerant user interface will find ReactJS as the go-to choice. The component creation aspect of this library allows developers to efficiently perform unit testing, making sure no system crashes occur. Code reuse enables for curtailing time performing redundant tests. Adding such tests improves standards in code quality; hence, platform stability.

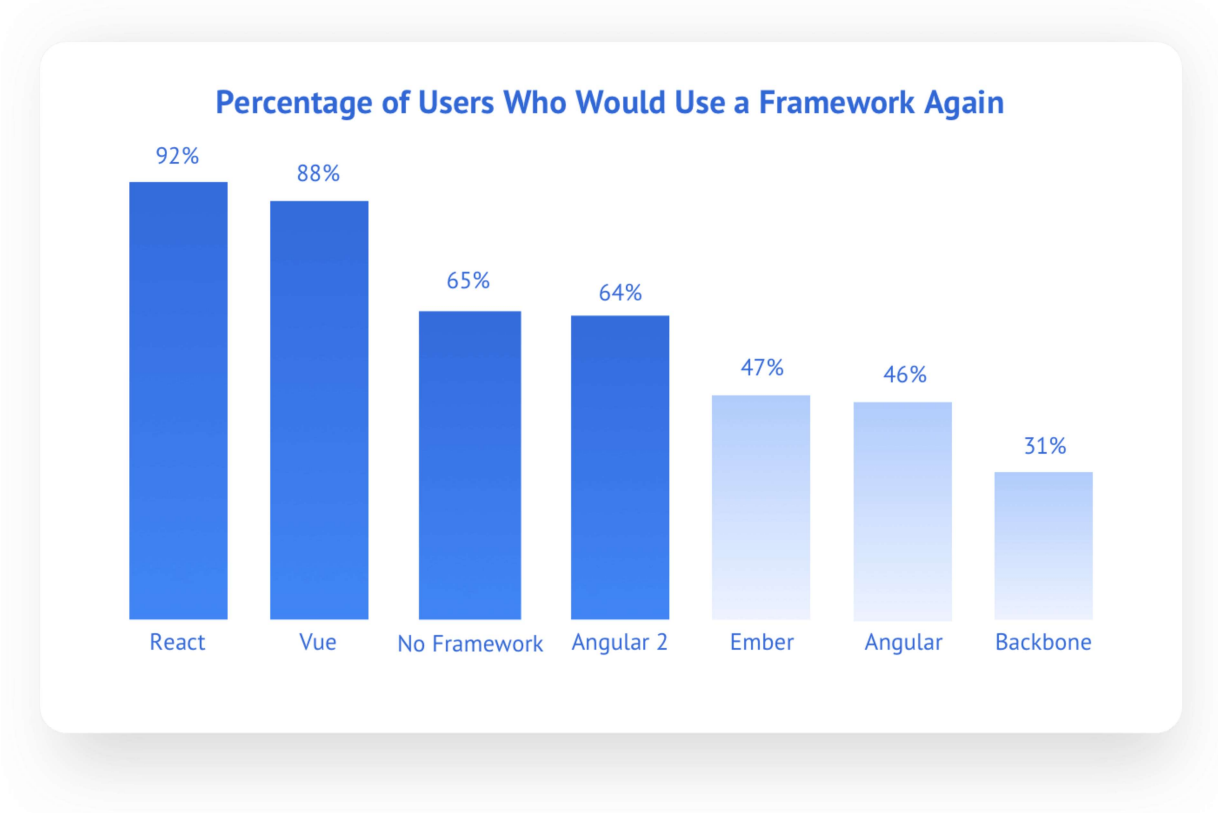
Uses of ReactJS and its popularity

While other technologies such as Angular were available when Facebook developed ReactJS, most developers were forced to do a lot of coding. Developers using other frameworks have the challenge of having to rework on most codes even when crafting components that changed frequently.

What they wanted was a framework that could allow them to break down complex components and reuse the codes to complete their projects faster. ReactJS provided the solution that developers were looking for. It uses JSX (a unique syntax that allows HTML quotes as well as HTML tag syntax application for rendering specific subcomponents) This is very helpful in promoting the construction of machine-readable codes and at the same time compounding components into a single-time verifiable file.

Today, ReactJS has become highly popular because of its extra simplicity and flexibility. Many people are even referring to it as the future of web development. It is estimated that more than 1,300 developers and over 94,000 sites utilize ReactJS.

Part of this huge popularity comes from the fact that top corporations such as Facebook, PayPal, Uber, Instagram, and Airbnb use it to solve user interface related issues. This credibility has drawn a lot of people to the framework.



Node.js

Node.js came into existence when the original developers of JavaScript extended it from something you could only run in the browser to something you could run on your machine as a standalone application.

Now you can do much more with JavaScript than just making websites interactive.

JavaScript now has the capability to do things that other scripting languages like Python can do.

Both your browser JavaScript and Node.js run on the V8 JavaScript runtime engine. This engine takes your JavaScript code and converts it into faster machine code. Machine code is a low-level code that the computer can run without needing to first interpret it.

Open Source

Node.js is an open-source platform. It means that the copyright holder has given various rights of studying, editing and distributing the software to anyone for any purpose.

High Scalability

Since it uses an event mechanism, Node.js is highly scalable and helps the server in a non-blocking response.

Simple and Fast

As Node.js is built on Google Chrome's V8 JavaScript engine, its libraries are highly advanced and hence able to run the code at a faster speed.

No Buffering

Node.js is blessed with a special feature and that is, it does not buffer any data.

Single-Threaded

As it uses a process of event looping, Node.js is able to follow the single-threaded model. It helps a single user to handle more than one requests.

Asynchronous

Node.js has asynchronous libraries. It is quite helpful as Node.js servers need not wait for an API to send the response and move on to the next API.

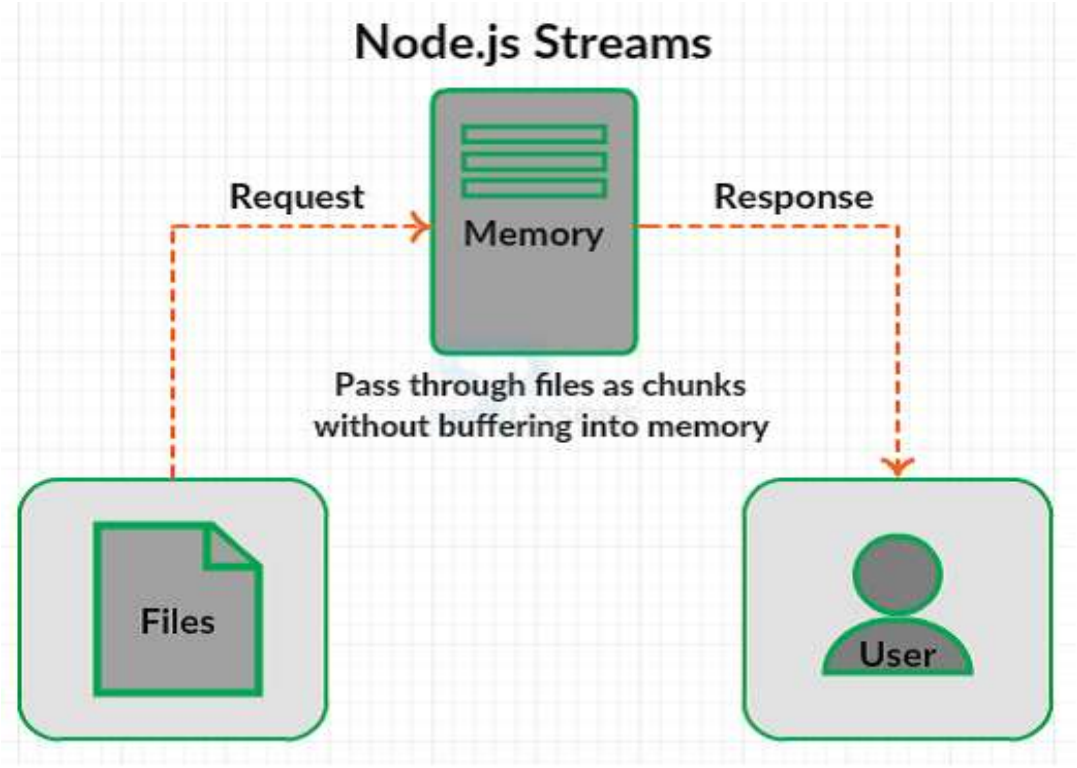
Cross-Platform

Node.js can be easily created and deployed on several platforms like Windows, Mac, and Linux.

Advantages of Node.Js

Faster suite

Since Node.js operates on Google's V8 engine, it is quite faster as compared to other technologies. Google's V8 engine is the fastest javascript engine right now.



Moreover, another reason for Nodejs to fast relies on the fact that Node.js uses NPM (Node Package Manager). NPM is an online repository that consists of several useful libraries and tools.

Easy to Learn

Anyone with little knowledge of JavaScript can easily learn to code in Node.js. It is because Node.js is a framework of JavaScript.
And when we talk about the learning curve of Node.js, it is quite shallow and it is less complicated to learn and use several libraries and toolkits available with it.

Caching
Node.js permits the functionality of caching of a single module. Ultimately, you need not re-execute the code while a request for the first module is created.

Data Streaming

In Node.js, Http requests and responses are treated as two different modules.
Hence, they are treated as two separate data streams.

So, whenever these files are loaded, the overall processing time gets reduced resulting in high efficiency. Ultimately, you can stream audios and videos without waiting for a long time.

Hosting

Node.js applications can be deployed easily. The cause lies in access to several hosting platforms such as PaaS (Platform as a Service) and Heroku.

Single Programming Language

Node.js can create web applications for front end as well as the backend. Because of this, the deployment of web applications has become a lot easier as most of the web browsers support JavaScript.

Real-time web applications

Node.js creates applications that have faster synchronization and reduced Http overload. This is the main reason behind the fact that developers prefer Node.js to create web-based chat and gaming applications.

Easy to Scale

Web applications that are developed using Node.js are easy to scale. Moreover, with Node.js, it is quite simple to add extra resources to it at the time of scaling an app.

More on Node.js

Node.Js Modules represents several functionalities that are grouped into single or more than one JS files. All these modules have a unique context and do not interfere with the scope of other modules.

These modules permit the code reusability and increase the ease of usage. Three modules that Nodejs offers:

Core Modules
Local Modules
Third-party modules

Core Modules

Node.js is a lightweight framework, code modules group absolute minimum functionalities. Generally, these modules get loaded just after the initiation of the Node process. The only thing to be done is to import all these core modules in order to use them in the code.

Core Module	Description
http	Contains classes, methods, and events required to create Node.js HTTP server
url	Contains methods for URL resolution and parsing in Node
querystring	Contains methods to deal with a query string of Node
path	Contains methods to deal with file paths
fs	Contains classes, methods, and events to work with file I/O
util	Contains utility functions that can be useful for programmers

Local Modules

Local modules are created locally by the user or dedicated software developer. All such modules may have several functionalities grouped into different files and folders. And all these can be distributed in the Nodejs community with the help of Node Package Manager.

External Modules

You can easily use these modules by downloading them through Node Package Manager. Moreover, generally these modules are developed by other developers and anyone can use them at free of cost.

Result Analysis:

We are checking the results on real time video using webcam as well as on pre-downloaded videos. We are getting good results but not as much we are expecting.

Secondly, Web platform that we made is working perfectly and is in sync with the backend of face detection properly.

We have trained our neural network on the FER2013 dataset. Both training and evaluation operations would be handled with FER2013 dataset. Compressed version of the dataset takes 92 MB space whereas uncompressed version takes 295 MB space. There are 28K training and 3K testing images in the dataset. Each image was stored as 48×48 pixel. The pure dataset consists of image pixels (48×48=2304 values) and emotion of each image.
Emotions are as: Disgust, Anger, Fear, Sadness, Happiness, Surprise and Contempt.

We got upto 65%accuracy on the facial expression recognition model on the test dataset of FER2013 which is quite good and we are continuously working on the model to make it more accurate and we are sure that we will come up with great model at the end.

For Face Identification, We are implementing VGG16 model along with a classifier model which label the identity of the person corresponding to its face features.

We have recognised the faces of different celebrities. As we have low size dataset so it is not showing that good results but it still good is we see from the level we have trained the model.

References

1. https://hackernoon.com/why-node-js-features-and-advantages-1k5c30cu
2. https://reactjs.org/docs/getting-started.html
3. https://expressjs.com/en/api.html
4. https://docs.mongodb.com/compass/master/
5. https://nodejs.org/en/docs/
6. https://www.udemy.com/course/react-the-complete-guide-incl-redux/learn/lecture/8156742?start=0#content
7. https://www.udemy.com/course/nodejs-express-mongodb-bootcamp/?start=0#overview
8. https://www.freecodecamp.org/news/what-exactly-is-node-js-ae36e97449f5/
9. https://medium.com/@thinkwik/why-reactjs-is-gaining-so-much-popularity-these-days-c3aa686ec0b3
10. https://docs.mongodb.com/compass/master/instance/
11. https://cloud.mongodb.com/v2/5dd6977eff7a253756c74858#clusters
12. https://towardsdatascience.com/build-your-own-convolution-neural-network-in-5-mins-4217c2cf964f
13. https://hackernoon.com/building-a-facial-recognition-pipeline-with-deep-learning-in-tensorflow-66e7645015b8