Object Identifier Using Image Analysis

Priya Mule[1], Yash Rane[2], Samyak Panchbhai[3], Shubham Khairnar[4]

Department of EXTC, Ramrao Adik Institute of Technology (Nerul): DY Patil Deemed to be University, Navi Mumbai, India.

Article Info
Volume 83
Page Number: 102- 107
Publication Issue:
November/December 2020

Abstract

A web application is being developed which can be used to identify an image, and thus detect the object present in it. A number of open - source frameworks are being used for the successful implementation of this project. The total cost incurred in this project is nil and it showcases the power of artificial intelligence in web applications.

Key Words: Object identification, web application

Article History
Article Received*: 25 October 2020*
Revised*: 22 November 2020*
Accepted*: 10 December 2020*
Publication*: 31 December 2020*

1. Introduction and explanation:

Object recognition is a modern computing technique that helps in the detection and identification of objects within a given frame of reference. This recognition model being developed is independently capable of identifying numerous different objects, and can prove to be effective as an all- purpose solution for a variety of visual recognition needs.

To do this, we have used a number of open-frameworks inthe designing and building of the website. We like to call these the major "pillars" or components that hold our entire application firmly on its feet.

1.1Front-end

The first component is ReactJS, which is an open-source JavaScript library for building user interfaces and UI- components. It has been used as the base for the developmentof our application. React helps to access multiple pages from a single page without a constant necessity to re-enter the required data. ReactJS is popularly used in major applications such as Facebook and Instagram. We have used ReactJS for the front-end of the web application.

1.2 Database

The second major component is PostgreSQL, also known as "Postgres", which is a free and open-source relational database management system that emphasizes extensibility and SQL compliance. PostgreSQL has been used to keep a track of user entities in a single database.

An alternative to PostgreSQL we also considered was Firestore. Firestore is Firebase's newest database for application development, developed by Google. This is a non-relational database. Non-relational databases are generally faster because a query does not have to view several tables, leading to rapid authentication of incoming users. However, inspite of this, relational databases have a distinct advantage of having an easier access to data, while also being more flexible to the dataset as a whole.

1.3 Server

The third component of our application is NodeJS. This is a cross-platform, back-end JavaScript runtime environment, which we have used as our server. NodeJS helps to keeptrack of the end-points and helps us to connect our relational database to the front-end of the web application. Along with NodeJS we have also used a small library called ExpressJS which helps making the transactions between the frontend and server easier.

1.4 API model

The fourth major pillar, on which our website stands is theAPI model. Every API is of course unique in itself, having a different model and functionality. We considered the possibility of both a pre-built as well as a self-trained detection model, however owing to the fact that pre-trained models have an extensive database at their disposal[1], we decided to go ahead with the pre-trained API model.
For implementing the object detection part we have usedan API called 'Clarifai API'. Clarifai API has quite a few models under it's hood of which we have implemented a model called 'general model'. The general model is trained with a dataset consisting of over thousands and lakhs of images. It is so vivid that it claims to predict up to 11,000 concepts which is amazing in itself and hence we get properpredictions.

2. Working:

2.1 Initialization

2.1.1 Initialization of ReactJS

For this process we have to install an IDE or code-editor to be able to write the code and modify the required files. Wehave used VS code editor by Microsoft in this project. You also need to install Node Package Manager (NPM) which can be installed while installing NodeJS. Once NPM is installed successfully, go to your terminal or bash and enter the command 'npx create-react-app my-app'. This command starts downloading and installing the required files we needin order to initialise and run a react application. Once that installation process is completed you will see a few files in the my-app folder. These are the basic react files which ship after react installation. The "src" folder consists of the main files which we will modify with our code.
Here, Index.js is the parent or the upper most element which executes the code. App.js is a primary child to index.js and is used to contain and execute all the components required in the code. React is based on component based design system hence we have made and rendered different components to fit our needs. We also installed a few librariesor dependencies to make the frontend code more functional and interactive. Here is the list of libraries that we haveinstalled for the front-end:

- Particle js
- Tilt js
- Clarifai
- Tachyons

All the libraries mentioned above can be downloadedusing NPM in a similar manner as React.

2.1.2 Initialization of NodeJS

Initializing NodeJS is comparatively simpler. Once NodeJS is installed in your system, you just have to run the command 'npm init' which will download the required modules also known as 'node modules' and a package.json file which allows us to run the code. Now we just have to create a new file and other middleware and controller files if necessary.
Below is a list of libraries and dependencies that we have installed in order to make server transactions easier:

- ExpressJS
- Knex
- Cors
- Bcrypt
- Nodemon

2.1.3 *Initialization of PostgreSQL*

For this process we need to install PostgreSQL from their website and setup the environment variables for proper functioning. Initially we also install PG admin which is a GUI for Postgre and helps us to keep a track on everything about the database we create in a simple manner.
Post installation, from the terminal we can run the command ‘CREATE DATABASE app’ which creates a database named app in PostgreSQL. We use the knex library to handle the requests to and from the database.

2.1.4 *Initialization of Clarifai API*

Initialization of the API is pretty simple. You have to create an account at Clarifai. After successful acount creationyou get redirected to your dashboard where you have tocreate an application, give it a name and once the application is setup, you are given an API key[2]. API key is unique to every user and hence its important to keep it safe and it is necessary to provide the key in our main application while sending POST requests to an API model. Once you get the API key, just select the api model you want to use, in our case it’s the ‘general model’. Go to the docs of that model and you’ll get required commands and function that we have to mention in our main application in order to make a requestto the API. Also the API key is used to determine the numberof requests an user or his application has made to the servers. For develpers, Clarifai API offers a free plan which allows around 5000 requests made for one month and hence if you have selected the free plan and the requests exceed the limit of 5000, the server will throw an error.

2.2 Working at Front-end

In the context of front-end, we have made two things, a landing page with a mindset of getting the project SEO compaitible and increase its ranking. And the main application which is made using ReactJS. Landing page is implemented using HTML and CSS and it consists of a link to the main application, project information and some information about the contributors and creators of this project.
When it comes to the main application, the user is greetedwith a signin page by default. The signin page has the username and password fields and a link to the register page. If the user is an existing user in our application, he can just input his credentials and signin else if he’s a new user, he cango to the register page and register with his details.
Upon successful authentication, the user is redirected to the main page of our application and is greeted with a beautiful UI. The header now dynamically changes to include a signout link. You can see a logo with an added tilt effect so when an user hovers over it, he can experience a smooth tilt animation. Below that is a greeting message and the username for the user who has logged in is fetched from the database and displayed in the message.
Now, in the input box a user has to input the imageaddress of the image he wants to scan, analyse or see the predictions for and once he clicks on detect, the image he wants to scan appears right below with a table showing the predictions and its percentage. Once an user clicks on detect with an iage address, that address is sent to the Clarifai servers using POST request and the response for that request is fetched using the predict function by Clarifai and displayed in the form of table using DOM manipulation.[3]

2.3 Working at backend

The backend consists of a server made using NodeJS anda database implemented using PostgreSQL. The server has the following end points

- Signin
- Register
- Image
- Profile

Whenever a user inputs his credentials and signs in that input is sent to the signin endpoint and then the server interacts with the database and checks whether the user

with these credentials exists and if the credentials are right. If bothcases return true, the user is redirected to the main page andif one of the cases turns false, an error is thrown.
On the register page, when an user inputs his desiredcredentials and clicks on register, the data is sent to theregister end point and then it is further stored in our database.If the user with those credentials already exists in our database or if the spaces are left blank, an error is thrown.
In both the signin and register endpoints, the password is not stored as it is. It is encrypted using MD5 algorithm provided by the bcrypt library before getting included in the request body. That means, even in a middle man attack or any cyber attack on the website, even if the hacker gains access to the email address, he won't be able to access the password.
The image end point is used to keep the entries or number of images an user has scanned updated. So whenever an user detects an image successfully, the entry count is incremented by 1 which is also displayed on the front-end of the website.
The database mainly consists of two tables, login andusers. As the name suggests, the login table consists of an id, a password hash and the login email. The users table consists of an id, username, email, number of entries or images scanned and lastly a timestamp of when the user first registered at our site.

2.4 Hosting

The hosting part is implemented using Heroku. To use heroku, we have to install the heroku-cli which is a commandline interface allowing us to use heroku commands to deploy our site. To get started, one has to create an account at Heroku. After logging in we can see a bunch of applications which heroku can host on its servers.
In order to host the website with ease, we have to host the code files on github. After uploading the code files, go to the terminal and type the command 'heroku create'. This will create a heroku app to which we can further host our application. All the steps regarding the initialization and deployment of an app on heroku can be found in their documentation.
We have used heroku to host the frontend as well as the backend of our website. So the react app, the node server and the postgre database, all three are hosted on separate heroku servers/computers and all of them work in unision to provide a smooth and seamless experience to the end user.

Motivation, Scope and Applications

The main motivation behind the creation of this web application was to do a practical implementation of our knowledge in the field of computer science, artificial intelligence and machine learning, The total cost of building the application is zero, which we believer is a great victory initself. The project has been built from scratch and hosted on all open-source platforms, leading us to successfully achieve the feat of making a full-stack web application without incurring any charges.
For the purpose of this paper, the scope of the project has been confined to the website. However, the idea offers innumerable scopes for application on a wider scale. The driver-less cars we see around us today make use of an advanced form of the same technology to detect any obstacles whatsoever, that might be present on the road. An implementation of this technology can also be found in modern smartphones with AI-enabled cameras. One just needs to point at an object, the camera scans it, and churns out an accurate result of the image being presented. The technology demonstrated in this paper can also be used in digital marketing and creative campaigns, with visual search giving an effective medium to optimise searching algorithms for particular products that potential buyers might be looking for.[4]

References

[1] Laya K. Roy, Reshma K. V., "A Novel Method of Object Identification and Tagging Using Speeded-Up Robust Feature" 2018 2nd International Conference on Inventive

Communication and Computational Technlogies, ISBN:978- 1-5386-1975-9/ ©2018 IEEE

[2] Yuyin Sun, Liefeng Bo, Dieter Fox, “Attribute based Object Identification”, 2013 IEEE International Conference on Robotics and Automation, ISBN: 978-1-4673-5643-5/
©2013 IEEE

[3] Sheila Tejada, Craig A. Knolblock, Steven Minton, “Learning Object Identification Rules for Image Integration”, in Information Systems journal, vol. 26, issue 8, December 2011, pp. 607-633

[4] Pooja Agrawal, Teena Sharma, Nishchal K. Verma, “Supervised approach for object identification using speeded up robust features”, in International Journal of Advanced Intelligence Paradigms, vol. 15, issue 2, February 2020

object web app

ORIGINALITY REPORT

2%

SIMILARITY INDEX

PRIMARY SOURCES

1 covid-19.hscni.net 1% Internet 19 words —

2 www.thirdrocktechkno.com 1% Internet 18 words —

EXCLUDE QUOTES	OFF	EXCLUDE MATCHES	< 5 WORDS
EXCLUDE BIBLIOGRAPHY	ON		