PerceptiveAnalytics

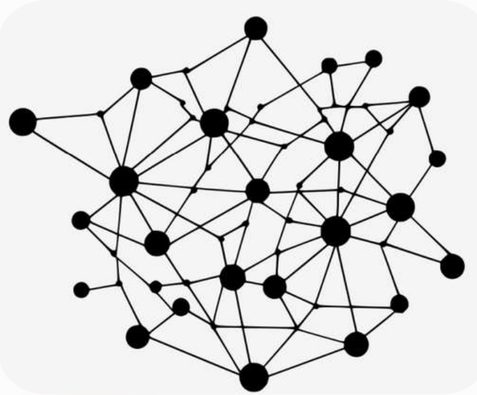
TEAM-33

RELEASE-2

Computer vision application for real-time multi-modal product detection

INTRODUCTION TO PRODUCT DETECTION

This multimodal product detection algorithm, identifies products from both static images and live video feeds.



In today's fast-paced world, efficient product detection holds immense significance across diverse industries.

Leveraging computer vision (CV) algorithms, empowered by deep neural networks, offers a promising solution to this challenge.

Target Market

Who are the customers we want to cater to?

Clients of Perceptive Analytics

A typical individual who is interested in basic functionalities and lacks extensive technical expertise

Retail Stores

Admin of stores, with a little knowledge of software but are handy in using applications on mobile devices.

Delivery System Companies

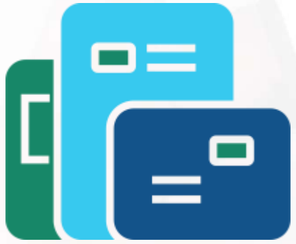
Officials who keep a check at the quantity delivered by delivery guy, who is handy with mobile apps.

Inventory Management

Helps in keeping the stock updates and need for restocking when required.

Recap of R1

Explored various frameworks and cloud services options



Mockup design of the entire application made on Figma

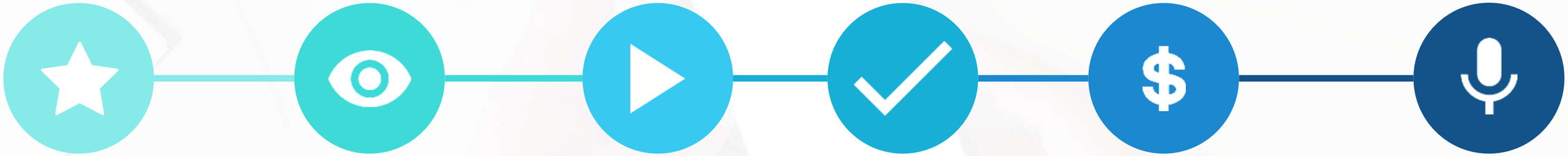
UI design implemented and frontend code for most pages completed

Backend for image processing and upload functionality integrated

R2

ROADMAP FOR R2

Transitioning between R1 and R2



R1

Basic structure set-up and UI implemented, in addition with Image processing

UI

Improvements made in UI as per client's request

VIDEO

Video processing implemented

TESTING

Detection Algorithm is enhanced and tested again

DEPLOYMENT

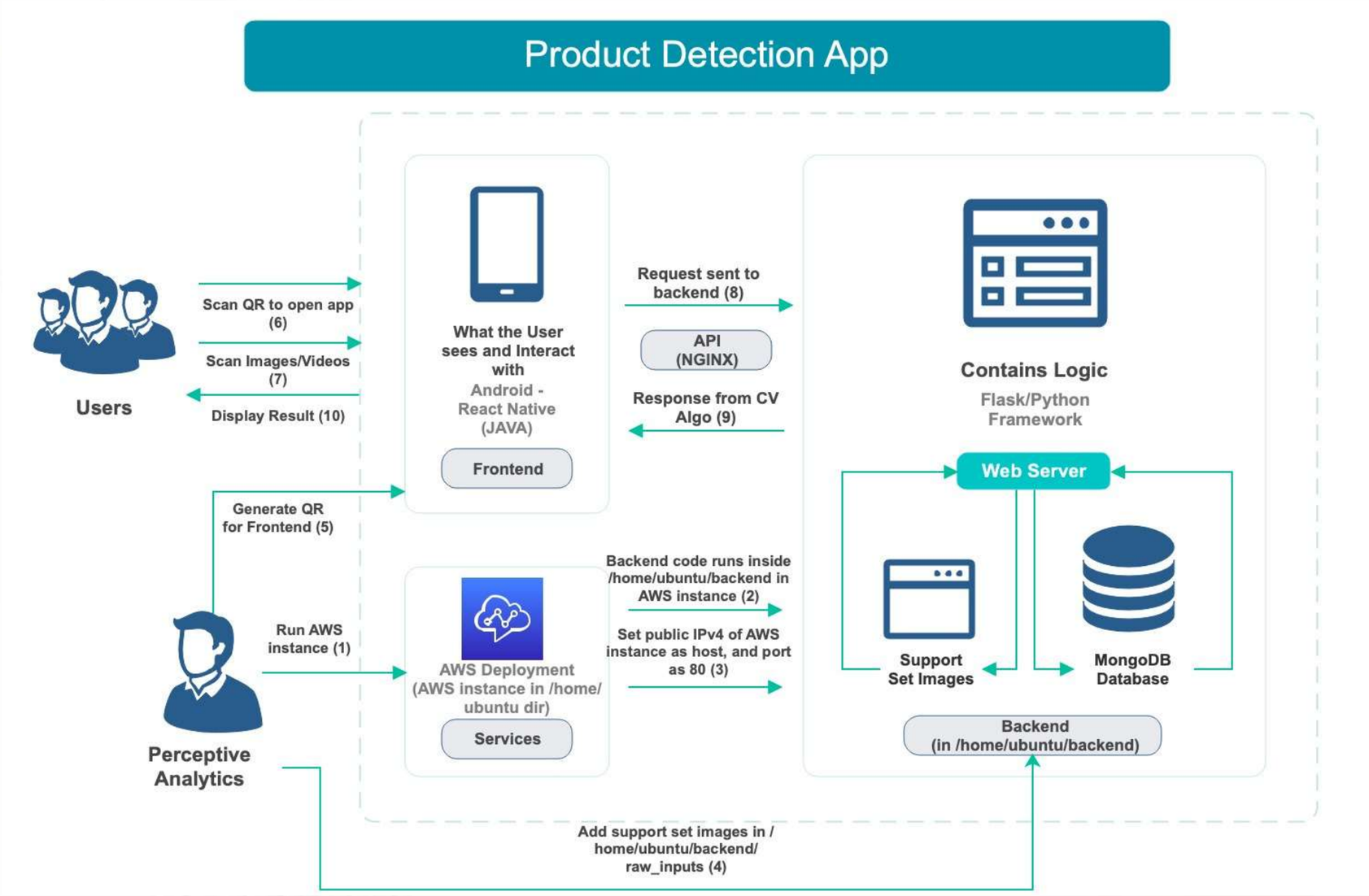
Backend deployed in AWS

DOCUMENTATION

Comprehensive documentation and KT videos for seamless understanding

APP ARCHITECTURE

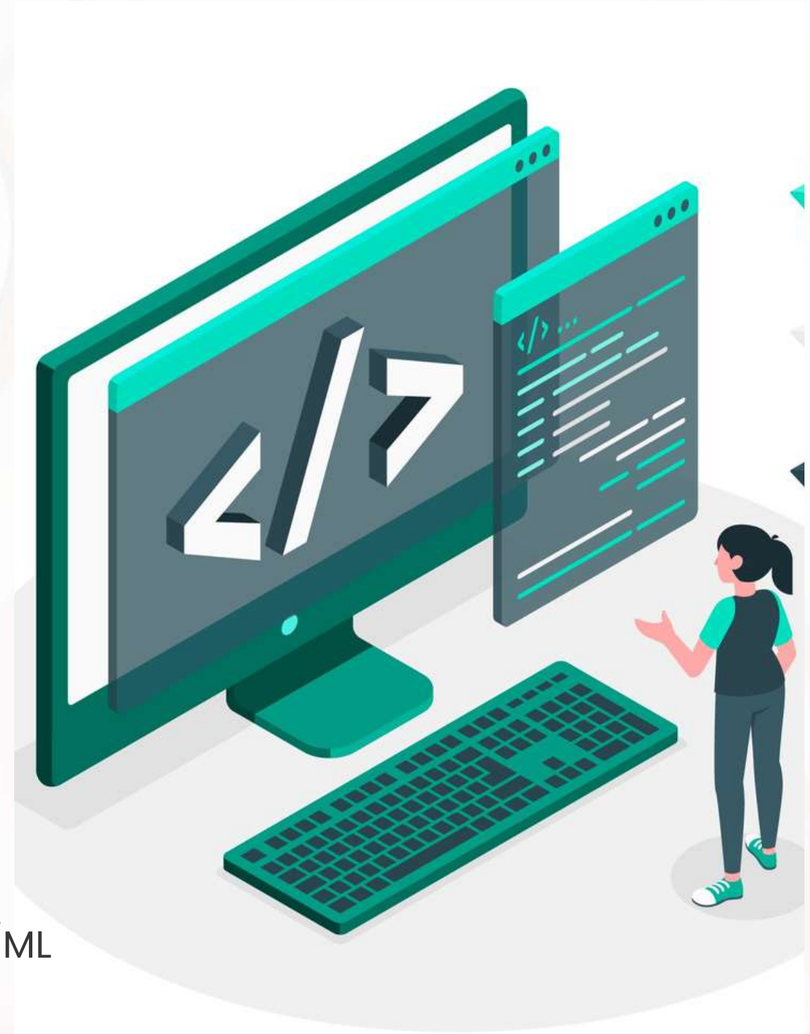
Top level overview of app deployed on AWS



Tech Stack

Backend

- Flask
- AWS
- NGINX



Frontend

- React-Native
- Node.js
- Expo

The backend choices provides global infrastructure and compatibility with AI/ML Libraries. The framework is appropriate given low-currency and light weight nature of app

The frontend choices are made keeping in mind use case and user load with support for third party tools for better feature implementation

Backend Deployment

AWS Elastic Cloud Compute (EC2) On-Demand High Computing Power (c5.4xlarge) instance

EC2 On-Demand Instance

1. Flexibility with choices of multiple instance types, software packages, instance storages, and operating systems.
2. Elastic Web-Scale Computing and Complete Control over root of instance.
3. Converts your large fixed costs into smaller variable costs, saving you from planning and managing costs.

APPS WITH **SHORT-TERM, SPIKY, OR UNPREDICTABLE WORKLOADS** THAT CAN'T BE INTERRUPTED, OR **DEPLOYED FOR INITIAL PHASES OF TESTING,** HENCE EC2 ON-DEMAND IS BEST SUITED FOR OUR TARGET MARKET

Assisting Frameworks

Instance Specs: Ubuntu 22.04 AMI, with HTTP and SSH inbound request security group, and 20 GB EBS root volume

NGINX

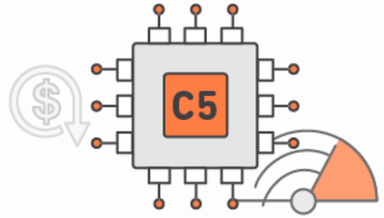
- Open-source web server software used for reverse proxy, load balancing, and caching.
- Offers scalability and the ability to handle concurrent requests.
- Speeds up performance by routing traffic to web servers in a way that improves speed
- Reduces the waiting time to load a website.

Gunicorn

- Python WSGI HTTP Server
- Easy to configure and easy to maintain
- Compatible with various web frameworks, simply implemented, light on server resource usage, and fairly speedy.

Elastic IPv4

- Public static IPv4 address which is reachable from the Internet.
- Can be quickly and easily remapped to different instances within your account.
- In the event of an instance failure, you can remap the it to a standby instance, minimizing downtime and improving disaster recovery capabilities.



C5 Instance

- Deliver cost-effective high performance at a low price per compute ratio
- Very popular for running compute-heavy workloads like batch processing, high-performance computing, machine/deep learning inference etc.
- C5 has more memory and higher EBS bandwidth across all instance sizes.

Major

Features

Image/Video Viewer

- Previewing scanned input
- Observing Detections from Algorithm
- Frame by Frame for Video Analysis

Qualitites

★ **Extendibility of use case for order placement with one-click on tick mark**

Image/Video Viewer

The image/Video viewer represents scanned input , for verification and preview. On clicking, one can also observe detection frames, indicating towards items detected by algorithm.

Detections

It displays list of items detected by algorithm with its quantity.

Custom Quantity

The user on scanning single piece can edit the quantity as per their requirement.

Major

Features

Click/Upload Images and Videos

- Click Photo
- Click Video
- Upload from Gallery

Camera button

Long Press for Photo, single clicks for video start/stop

Upload button

To upload an image/video from galle

Qualitites

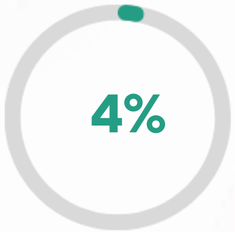
★ **Minimalistic design with all features at one place for easier accessibility**

Demonstration

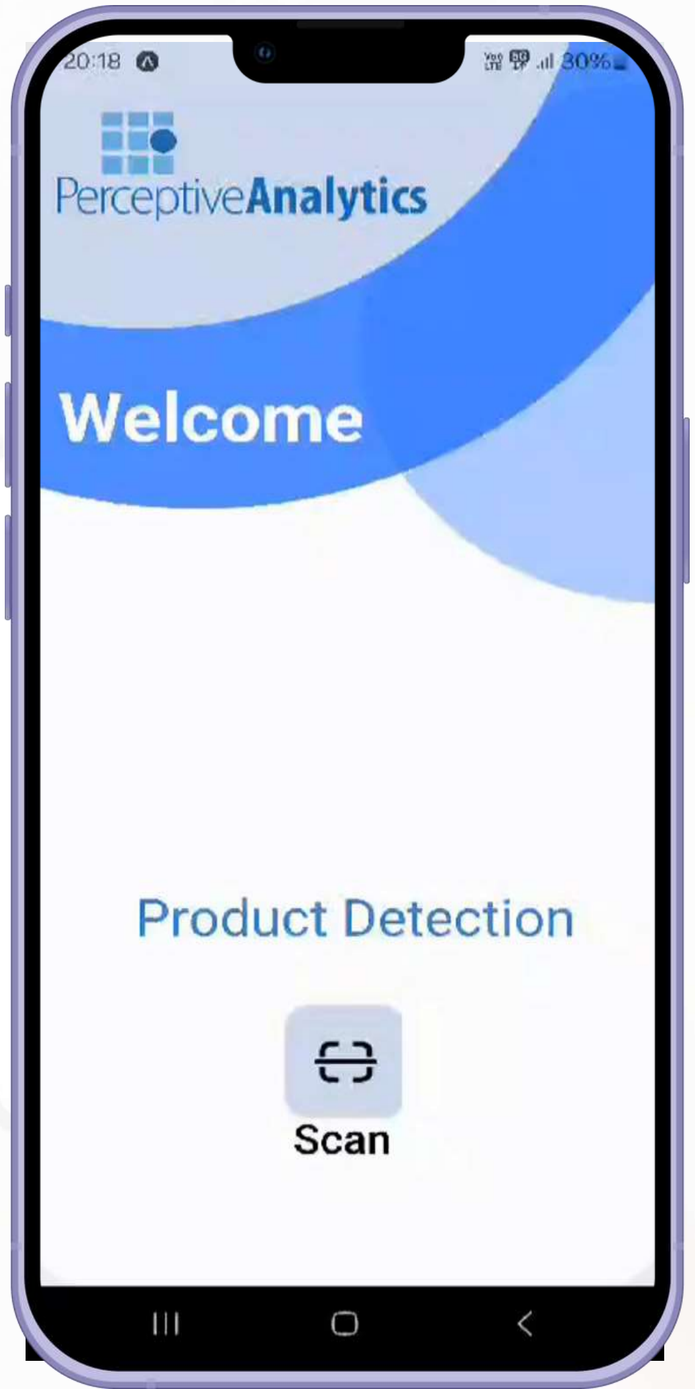
Performance: On standard run of video(5 frames), image and upload, the following metrics are generated

Processing Time

Start-Up Time



★ **The above performance is on CPU equivalent to i7**



Model
Min SDK 21.0

Pre-Requisite

- Steady Online Connection
- Camera

Summarising Deliverables

UI DESIGN

Easy to use, self explanatory and interactive interface

ROBUST APP ARCHITECTURE

Based on the current framework(s) of the app that the codebase is using, deprecations are removed and updated to compatible versions.

DEPLOYMENT

Deploying the app on AWS for enhanced accessibility across devices, leveraging cloud computing for ease of use.

UPDATION OF SUPPORT SET

Easy and Quick update of support set images for wider recognition of objects and easier access to developers

Challenges Faced

Finding a video previewer

Not all libraries were compatible.
Used expo-av finally due to compatibility and extensive documentation for reference.

NGINX error with HTTP requests

404 NGINX error when request was sent through HTTP.
Allocation of a static elastic IPv4 address to ensure uniformity in requests sent to the instance.

Integration of new algorithm

The algorithm is black box to us thus, while updating algorithm, we faced difficulty to identify flow of input and output received considering modules imported by them.

Usage of deprecated frameworks

Upgraded the versions of frameworks and used new functions in place of the ones removed

Learnings

Cost Analysis

Various options were considered for cloud deployment services. AWS was agreed upon finally due to better value for cost. Experiencing negotiation tactics.

Importance of Documentation

Choice of libraries such as expo-camera and expo-av was made majorly due to easy understanding of their functioning as mentioned in the documentation. Inspiring the documentation phase of this project for further development in future.

Acknowledge Sudden Changes

Learning how to handle new or updated requests and requirements, and accommodating them in the current workflow.

Meet our Team

Aanvik Bhatnagar

Developer

For R2:

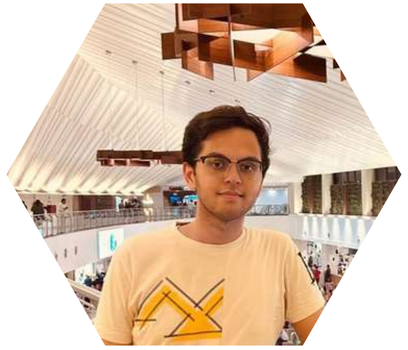
- AWS Deployment
- Video Processing
- Client Documentation: AWS Deployment Procedure
- Cost Analysis for Client

Rohan Naidu

Developer

For R2:

- Video Viewer UI
- Test Demo Video for Client
- R2 Presentation
- MoMs for Client meets



Chetan Mahipal

Developer

For R2:

- Video Processing
- Update Algo
- Client Documentation: Quick Startup
- Backend Testing
- Expo Deployment

Rohan S

Developer

For R2:

- AWS Instance Research
- Test Demo Video for Client
- R2 Presentation

Rohan Shridhar

Developer

For R2:

- Video Viewer UI
- Client Documentation: Workflow and App Architecture
- System Testing for UI
- R2 Presentation

THANK YOU

FOR INVESTING IN OUR PROJECT

April 2024